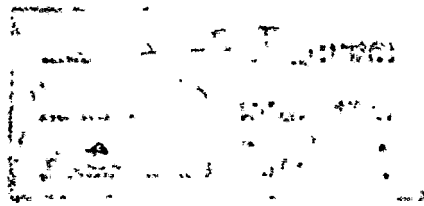




ระบบปรับสถานะ ประยุกต์ใช้ในงานกำจัดสัญญาณรบกวน  
ADAPTIVE SYSTEM, APPLICATION ON INTERFERENCE CANCELING



โดย  
นายพนรัตน์ สิงห์โตโรจน์  
นายสมยศ บำรุงสุข

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
ภาควิชาวิศวกรรมโทรคมนาคม  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2534

007709

ปริญญาโท ปีการศึกษา 2534

ภาควิชา วิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
เรื่อง ระบบปรับสถานะ ประยุกต์ใช้ในงานกำจัดสัญญาณรบกวน

ผู้จัดทำ

1. นพรัตน์ สิงห์โตโรจน์
2. สมยศ บำรุงสุข

..... อาจารย์ที่ปรึกษา

(ผศ. โมนัย ไกรฤกษ์)

## ระบบปรับสถานะ ประยุกต์ใช้ในงานกำจัดสัญญาณรบกวน

นาย นพรัตน์ สิงห์โตโรจน์

นาย สมยศ บำรุงสุข

อาจารย์ที่ปรึกษา

รศ. ดร. วิวัฒน์ กิรานนท์

ผศ. โมไนย ไกรฤกษ์

### บทคัดย่อ

โครงการนี้เป็นการศึกษา ระบบปรับสถานะ (adaptive system) ซึ่งมีหลักการ ปรับเปลี่ยน ทราบสเฟอรัฟังก์ชัน ของ นอนรีเคอร์ซีฟฟิลเตอร์ โดยปรับเปลี่ยนค่าถ่วงดุล หรือ ค่าสัมประสิทธิ์ของฟิลเตอร์ ด้วยการใช่วิธีค่าเฉลี่ยกำลังสองต่ำสุด (Least Mean Square algorithm) เพื่อหาจุดต่ำสุดในฟังก์ชันประเมิณผล และได้แสดงหลักการประยุกต์ใช้ระบบปรับสถานะ ในการกำจัดสัญญาณรบกวน และได้จำลองการทำงานของระบบกำจัดเสียงรบกวนปรับสถานะ ด้วยการใช้ออมพิวเตอร์ทำการคำนวณ LMS algorithm โดยได้แปรค่า สัญญาณและสัญญาณรบกวนที่ใช้ใน LMS algorithm แล้วพิจารณาผลกระทบ ของการแปรค่าสัญญาณต่าง ๆ และได้พิจารณาผลต่อการทำงาน ของการปรับค่าภายในขอบเขตจำกัดของ พารามิเตอร์ของการลู่เข้า

## Adaptive System, Application on Interference Canceling

Noprath Singtoroj

Somyos Bamrongsook

Advisors

Dr. Wiwat Kiranon

Monai Krairiksh

### Abstract

This paper is a study on Adaptive System, with application on interference canceling. The system has principal on adjusting nonrecursive filter's transfer function through adjustment of filter's weights or coefficients using Least-Mean-Square Error algorithm. The weights adjustments accomplish for moving system's performance to the lowest position on the performance surface. Concept of using adaptive system for interference canceling application and simulated results for such application using computer are included in this paper. With LMS algorithm, changing values of intened signals and interference results in interesting changes on output signal. Effects from changing convergence parameter in limited boundary are also observed.

## สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทรานสเฟอ์ฟังก์ชันของฟิลเตอร์	3
บทที่ 3 วงจรรวมเชิงเส้นแบบปรับสถานะ	6
3.1 สัญญาณอินพุท และ เวกเตอร์ของสัมประสิทธิ์	7
3.2 ผลตอบสนองที่ต้องการ และค่าความผิดพลาด	8
3.3 ฟังก์ชันประเมินผล	9
3.4 Least-Mean-Square Algorithm	11
บทที่ 4 การประยุกต์ใช้ระบบปรับสถานะในการกำจัดสัญญาณรบกวน	16
บทที่ 5 ผลการจำลอง (Simulated results)	20
บทที่ 6 วงจรเชื่อมต่อ	72
6.1 วงจรจำกัดขนาดสัญญาณ	73
6.2 วงจรเลื่อนสัญญาณ	74
บทที่ 7 สรุปผล	76
ภาคผนวก	
หนังสืออ้างอิง	
กิตติกรรมประกาศ	

บทที่ 1

บทนำ

1.1 กล่าวนำ

ระบบประมวลผลสัญญาณปรับสถานะ (adaptive signal processing) เป็นระบบประมวลผลสัญญาณดิจิทัล (digital signal processing) ที่มีโครงสร้างเป็นดิจิทัลฟิลเตอร์แบบแปรตามเวลา ซึ่งหลักการต่างจากดิจิทัลฟิลเตอร์โดยทั่วไป คือสัมประสิทธิ์หรือคูล่วงค่า (weights) ของระบบปรับสถานะมีค่าเปลี่ยนไปตามเวลา เพื่อให้ผลลัพธ์ (output) ของระบบคงสถานะตามลักษณะที่ต้องการ ไม่ว่าจะสภาวะแวดล้อม (environments) ที่มีผลต่อระบบจะเป็นอย่างไร หรือ คุณลักษณะ (characteristics) ของสัญญาณอินพุทเปลี่ยนไป ระบบปรับสถานะจะปรับค่าสัมประสิทธิ์ของตัวเองโดยอัตโนมัติ เพื่อให้คุณลักษณะของสัญญาณเอาท์พุทมีลักษณะคงที่ตลอดเวลา

การใช้งานระบบปรับสถานะนี้มีประโยชน์มาก เมื่อไม่สามารถควบคุมสภาวะแวดล้อมให้คงที่ ตามที่คิดไว้ในขั้นตอนออกแบบระบบ ซึ่งจะพบได้บ่อยครั้งในการออกแบบระบบจริง ๆ เพราะเป็นการยากที่จะควบคุม หรือ รู้ล่วงหน้าถึงคุณลักษณะของสภาวะแวดล้อม ในขณะที่ระบบกำลังทำงานได้ ดังนั้นการที่ระบบปรับสถานะสัญญาณสามารถปรับเปลี่ยนตัวเอง ให้สามารถทำงานได้ไม่ว่าจะอยู่ในสภาวะแวดล้อมแบบใด หรือสภาวะแวดล้อมเปลี่ยนแปลงตลอดเวลา (nonstationary) จึงเป็นข้อได้เปรียบของระบบนี้เหนือกว่า ดิจิทัลฟิลเตอร์ แบบธรรมดา

โครงการนี้ได้แสดงการประยุกต์ระบบปรับสถานะ มาใช้ในงานกำจัดสัญญาณรบกวน (interference canceling) โดยใช้สัญญาณรบกวนเป็นสภาวะแวดล้อมที่ควบคุมไม่ได้และใช้ระบบปรับสถานะ กำจัดสัญญาณรบกวนเหล่านั้นออก ซึ่งได้จำลองการกำจัดสัญญาณรบกวนแบบต่าง ๆ ทั้งสัญญาณรบกวนแบบมีคาบเวลา และไม่มีคาบเวลา

1.2 จุดประสงค์ของโครงการนี้

จากที่กล่าวมาข้างต้น เป็นการแนะนำหลักการ และข้อดีของการกำจัดสัญญาณรบกวน ด้วยระบบปรับสถานะ ซึ่งโครงการนี้ เป็นการศึกษา และประยุกต์ใช้เบื้องต้น เพื่อเป็นแนวทางที่จะพัฒนาระบบปรับสถานะ ให้มาประยุกต์ใช้กับงานอื่น ๆ ในทางโทรคมนาคมได้ต่อไป

### 1.3 โครงประกอบของรายงาน

รายงานประกอบโครงงานฉบับนี้แบ่งเนื้อหาออกเป็น 7 บท โดยมีบทนำ อยู่ในบทที่ 1 เป็นการกล่าวแนะนำข้อดี หลักการเบื้องต้นและวัตถุประสงค์ในการทำโครงงาน ส่วนรายละเอียดของบทต่าง ๆ อีก 6 บทมีดังต่อไปนี้

- บทที่ 2 กล่าวถึงทรานส์เฟอร์ฟังก์ชันของ ระบบสัญญาณดิจิทัล เป็นการแสดงว่า สัมประสิทธิ์ของฟิลเตอร์ มีความสัมพันธ์กับ ทรานส์เฟอร์ฟังก์ชัน
- บทที่ 3 กล่าวถึงวงจรรวมเชิงเส้นปรับสถานะ และการใช้ Least-Mean-Square algorithm ในระบบปรับสถานะ , แกรเดียนท์ของฟังก์ชัน, การประมาณค่า แกรเดียนท์
- บทที่ 4 กล่าวถึง การประยุกต์ใช้ระบบปรับสถานะ ในงานกำจัดสัญญาณรบกวน
- บทที่ 5 ผลการจำลองวงจรถัดสัญญาณรบกวน ด้วยสัญญาณ และสัญญาณรบกวนแบบต่าง ๆ
- บทที่ 6 กล่าวถึงวงจรเชื่อมต่อสัญญาณ เพื่อเป็นแนวทางสำหรับการใช้งานจริง
- บทที่ 7 เป็นการสรุปผล และวิจารณ์ถึงปัญหาที่ได้ พบ

## บทที่ 2

### ทรานสเฟอร์ฟังก์ชันของฟิลเตอร์

ดังที่ได้กล่าวมาแล้วในบทก่อนว่า ระบบปรับสถานะมีหลักการคือ ปรับค่าสัมประสิทธิ์ หรือ ค่าถ่วงดุลของดิจิตอลฟิลเตอร์ ในระหว่างการทำงาน ซึ่งการปรับค่าถ่วงดุลเหล่านี้ จะทำให้ทรานสเฟอร์ฟังก์ชันของฟิลเตอร์เปลี่ยนไป การที่ทรานสเฟอร์ฟังก์ชันของระบบใด ๆ เปลี่ยนไป จะทำให้เอาต์พุต ของระบบที่มีลักษณะอินพุตคงที่ เปลี่ยนไป แต่ระบบปรับสถานะจะกลับกัน คือ จะเปลี่ยนทรานสเฟอร์ฟังก์ชันเพื่อให้คงลักษณะเอาต์พุตไว้ ไม่ว่าจะลักษณะอินพุตจะเปลี่ยนไปอย่างไรก็ตาม

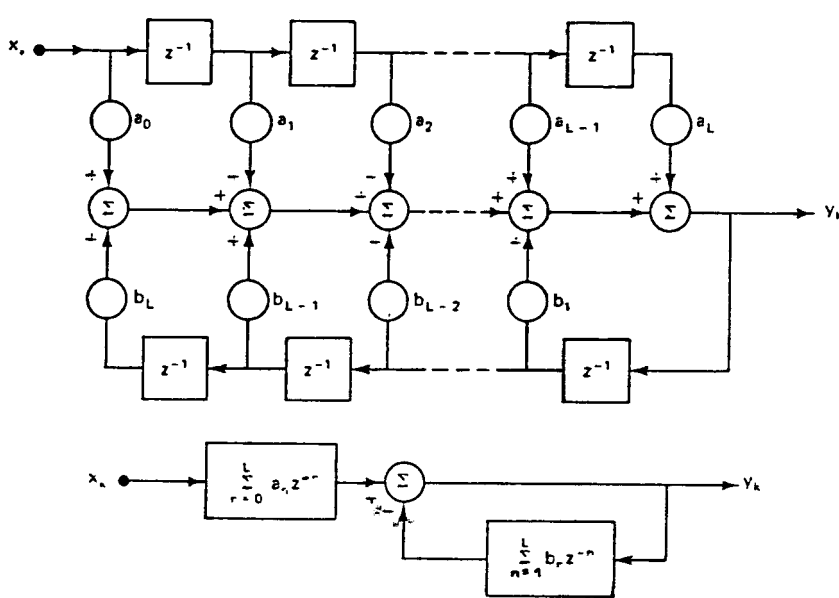
ในระบบปรับสถานะ ทั้งสัญญาณอินพุตและสัญญาณเอาต์พุต ต่างก็เป็นสัญญาณไม่ต่อเนื่องตามเวลา (discrete signal) จึงใช้ z-transform หาทรานสเฟอร์ฟังก์ชัน ซึ่งนิยามโดย z-transform ของสัญญาณเอาต์พุตหารด้วย z-transform ของสัญญาณอินพุต

$$\text{transfer function} = H(z) = Y(z)/X(z)$$

เมื่อ  $X(z)$  : z-transform ของสัญญาณอินพุต

$Y(z)$  : z-transform ของสัญญาณเอาต์พุต

โครงสร้างทั่วไปของดิจิตอลฟิลเตอร์ แสดงดังรูปที่ 2.1



รูปที่ 2.1 แสดงโครงสร้างของดิจิตอลฟิลเตอร์

สามารถนิยามสัญญาณอินพุต และสัญญาณเอาต์พุต เพื่อหาทรานสเฟอร์ ฟังก์ชันได้ดังนี้

- $X_k$  : สัญญาณอินพุต ที่เวลา  $t=k$  (เรียกว่า  $X$  ที่เวลา  $k$ )  
 $Y_k$  : สัญญาณเอาต์พุต ที่เวลา  $t=k$  (เรียกว่า  $Y$  ที่เวลา  $k$ )  
 $L$  : จำนวนครั้งของการหน่วงเวลา แต่ละครั้งจะหน่วงเวลาเท่ากับ  $k_n - k_{n-1}$  เรียก  $L$  ว่า ความยาวของฟิลเตอร์ (filter length)  
 $a_n$  : สัมประสิทธิ์ หรือ ค่าถ่วงดุลของฟิลเตอร์ ในทางเดินไปข้างหน้า (forward path) ที่คูณกับสัญญาณอินพุต ที่ได้หน่วงเวลามา  $n$  ช่วงเวลา  
 $b_n$  : สัมประสิทธิ์ หรือ ค่าถ่วงดุลของฟิลเตอร์ในทางเดินย้อนกลับ (feedback path) ที่คูณกับสัญญาณเอาต์พุต ที่ได้หน่วงเวลามา  $n$  ช่วงเวลา

จากรูปที่ 2.1 สามารถแสดงความสัมพันธ์ของสัญญาณเอาต์พุต ที่เวลา  $t=k$  ( $Y_k$ ) กับสัญญาณอินพุต ที่ เวลา  $t=k$  ( $X_k$ ) แสดงได้ดังสมการ

$$Y_k = \sum_{n=0}^L a_n X_{k-n} + \sum_{n=1}^L b_n Y_{k-n} \quad (2.1)$$

เทอมที่สองทางด้านขวามือของสมการที่ 2.1 เป็นส่วนที่เกิดจาก ทางเดินย้อนกลับ จะทำให้เอาต์พุต ( $Y_k$ ) ไม่เป็นศูนย์เมื่อไม่มี  $b_n$  ในที่เป็นศูนย์ ถึงแม้อินพุต ( $X_k$ ) จะเท่ากับศูนย์ ไปแล้วก็ตาม เรียกฟิลเตอร์ที่มีส่วนของทางเดินย้อนกลับว่า รีเคอร์ซีฟฟิลเตอร์ (recursive filter) และเรียกฟิลเตอร์ที่ไม่มีส่วนของทางเดินย้อนกลับว่า นอนรีเคอร์ซีฟฟิลเตอร์ (nonrecursive filter) ซึ่งมีรูปสมการ

$$Y_k = \sum_{n=0}^L a_n X_{k-n} \quad (2.2)$$

$a_n$  มีค่าเป็นศูนย์เมื่อ  $n < 0$  หรือ  $n > L$

เนื่องจากในขั้นนี้เป็นการศึกษาระบบปรับสถานะขึ้นเบื้องต้น เพื่อลดความยุ่งยาก จึงศึกษาเฉพาะระบบที่เป็น นอนรีเคอร์ซีฟฟิลเตอร์ เท่านั้น

หาทรานสเฟอ์ฟังก์ชันของฟิลเตอร์โดย หา z-transform ของสมการที่ 2.2

$$Y(z) = \sum_{n=-\infty}^{\infty} \sum_{k=0}^L a_n X_{k-n} z^{-k} \quad (2.3)$$

เนื่องจาก  $a_n$  เป็นศูนย์เมื่อ  $n < 0$  และ  $n > L$  สมการที่ 2.3 จึงสามารถเขียนใหม่ได้

$$Y(z) = \sum_{n=-\infty}^{\infty} a_n \sum_{k=-\infty}^{\infty} X_{k-n} z^{-k} \quad (2.4)$$

กำหนดให้  $m = k - n$  โดยที่  $m$  ยังมี limit ตั้งแต่  $-\infty$  ถึง  $\infty$

$$Y(z) = \sum_{n=-\infty}^{\infty} a_n z^{-n} \sum_{m=-\infty}^{\infty} X_m z^{-m} = A(z)X(z) \quad (2.5)$$

$$\begin{aligned} \text{transfer function : } H(z) &= Y(z)/X(z) \\ &= A(z)X(z)/X(z) \\ &= A(z) \end{aligned} \quad (2.6)$$

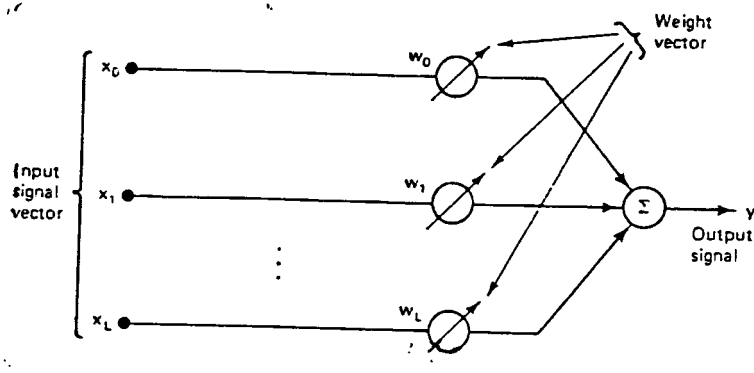
จากสมการที่ 2.6 เห็นได้ว่า ทรานสเฟอ์ฟังก์ชันของ นอนรีเคอร์-ซีฟฟิลเตอร์ คือค่า z-transform ของสัมประสิทธิ์ หรือ ค่าถ่วงดุลของฟิลเตอร์ ดังนั้น ระบบปรับสถานะ สามารถปรับทรานสเฟอ์ฟังก์ชัน ของตัวเอง เพื่อให้คุณลักษณะของเอาต์พุตมีค่าคงที่ ทำได้โดยปรับค่าสัมประสิทธิ์ หรือ ค่าถ่วงดุลของฟิลเตอร์นั่นเอง ในบทต่อไปจะอธิบายถึง การตัดสินใจเกี่ยวกับการปรับค่าถ่วงดุลเหล่านี้

### บทที่ 3

## วงจรรวมเชิงเส้นแบบปรับสถานะ (adaptive linear combiner)

ในบทที่แล้ว ได้แสดงให้เห็นแล้วว่า การที่จะปรับเปลี่ยน ทรานสเฟอร์ ฟังก์ชัน สามารถทำได้โดยการปรับเปลี่ยนสัมประสิทธิ์ของฟิลเตอร์ ในหัวข้อนี้จะเป็น การพิจารณาว่า ค่าสัมประสิทธิ์ที่ปรับเปลี่ยนไป ควรจะปรับเปลี่ยนไปอย่างไร เพิ่มขึ้น หรือลดลงเท่าไร ใช้อะไรเป็นตัวกำหนด

วงจรรวมเชิงเส้นแบบปรับสถานะ (adaptive linear combiner) เป็นวงจร นอนรีเคอร์ซีฟฟิลเตอร์ ที่แปรตามเวลา แสดงในรูปที่ 3.1 เป็นต้นแบบ ในการวิเคราะห์ระบบปรับสถานะที่ได้ทำการศึกษา



รูปที่ 3.1 รูปทั่วไปของวงจรรวมเชิงเส้นปรับสถานะ

เวกเตอร์ของสัญญาณอินพุต ประกอบไปด้วยอีลิเมนต์ภายในเวกเตอร์คือ  $x_0, x_1, \dots, x_L$  เวกเตอร์อินพุตจะสัมพันธ์กับเวกเตอร์ ของค่าถ่วงคูณ ที่มีอีลิเมนต์ในเวกเตอร์คือ  $w_0, w_1, \dots, w_L$  ด้วยการคูณเวกเตอร์ทั้งสองนี้ด้วยกัน จะได้ค่าออกมาเป็น ค่าสเกลล่าคือ เอาท์พุท  $y$  ขั้นตอนที่ระบบปรับค่าถ่วงคูณ เรียกว่า การปรับสถานะ และขณะที่เวลา  $t=k$  ใด ๆ ในระหว่างการปรับสถานะนี้ ระบบจะเป็นระบบเชิงเส้น เพราะค่าถ่วงคูณจะคงที่ ที่เวลานั้น ๆ ดังนั้นสัญญาณเอาท์พุทจะเป็น การรวมกันแบบเชิงเส้นของ อีลิเมนต์ของเวกเตอร์สัญญาณ แต่อย่างไรก็ตาม เมื่อ ค่าสัมประสิทธิ์อยู่ระหว่างการปรับสถานะ ค่าสัมประสิทธิ์นี้จะไม่คงที่ และระบบจะไม่เป็นเชิงเส้นในระหว่างการปรับสถานะ

### 3.1 สัญญาณอินพุต และ เวกเตอร์ของสัมประสิทธิ์

เวกเตอร์ของสัญญาณอินพุตในรูปที่ 3.1 สามารถมีความหมายได้สองอย่าง คือ

1. เป็นสัญญาณอินพุตที่เข้ามาพร้อม ๆ กันจากแหล่งกำเนิดต่าง ๆ กันจำนวน  $L+1$  แหล่ง

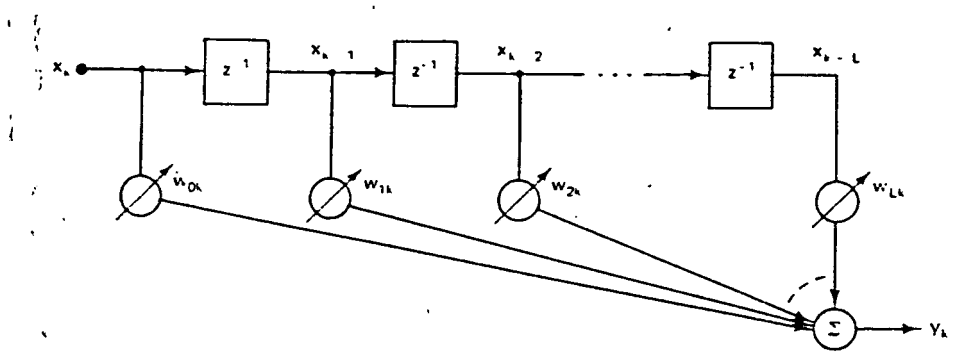
2. เป็นสัญญาณอินพุตจากแหล่งเดียวกัน ที่รับเข้ามาต่อ ๆ กัน  $L+1$  ครั้ง จะเรียกสัญญาณอินพุตของกรณีที่ 1 ว่าเป็นกรณี หลายอินพุต และจะเรียกอีกกรณีว่า กรณีอินพุตเดี่ยว นิยามเวกเตอร์อินพุตทั้งสองกรณีได้ดังสมการ

$$\text{หลายอินพุต} : X_k = [x_{0k} \ x_{1k} \ x_{2k} \ \dots \ x_{Lk}]^T \quad (3.1)$$

$$\text{อินพุตเดี่ยว} : X_k = [x_k \ x_{k-1} \ \dots \ x_{k-L}]^T \quad (3.2)$$

สัญลักษณ์ T (transpose) แสดงว่าทั้งสองกรณีสัญญาณอินพุตเป็นคอลัมน์เวกเตอร์ และ  $k$  เป็นดัชนีชี้เวลาที่เวลา  $t=k$  ซึ่งกรณีหลายอินพุต ทุกอิลิเมนต์ (สัญญาณจากหลายแหล่ง) ในเวกเตอร์ จะถูกเก็บข้อมูลที่เวลาเดียวกัน แต่สำหรับกรณีอินพุตเดี่ยว อิลิเมนต์ต่าง ๆ จะหมายถึงสัญญาณจากแหล่งเดียวกัน แต่ถูกหน่วงเวลาไปเท่ากับเวลาที่ใช้ในการอ่านสัญญาณแต่ละครั้ง

สำหรับกรณีอินพุตเดี่ยว วงจรจะประกอบไปด้วย วงจรรวมเชิงเส้นแบบปรับสถานะ และวงจรหน่วงหนึ่งหน่วยเวลา มาประกอบกัน ดังแสดงในรูปที่ 3.2



รูปที่ 3.2 วงจรรวมเชิงเส้นปรับสถานะแบบอินพุตเดี่ยว

โครงสร้างของฟิลเตอร์แบบในรูปที่ 3.2 เรียกว่า ฟิลเตอร์แบบขวาง (transversal filter) สังเกตตัวห้อย  $k$  ที่ได้เพิ่มเข้าไปที่สัมประสิทธิ์  $w$  เพื่อให้เด่นชัดว่า ฟิลเตอร์นี้แปรตามเวลา

หาความสัมพันธ์ระหว่างอินพุต และ เอาท์พุทของทั้งสองกรณี ได้ดัง  
สมการ

$$\text{อินพุตเดี่ยว} : y_k = \sum_{l=0}^L w_{lk} x_{k-l} \quad (3.3)$$

$$\text{หลายอินพุต} : y_k = \sum_{l=0}^L w_{lk} x_{lk} \quad (3.4)$$

โดยที่ นิยามเวกเตอร์ของค่าสัมประสิทธิ์ ดังนี้

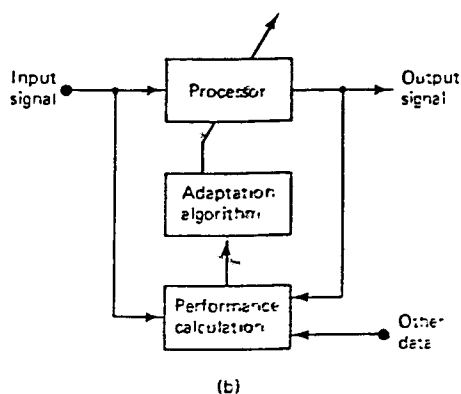
$$W_k = [w_{0k} \quad w_{1k} \quad \dots \quad w_{Lk}]^T \quad (3.5)$$

ดังนั้นจึงสามารถแสดงความสัมพันธ์ของ สัญญาณอินพุตกับสัญญาณเอาท์  
พุทในสมการที่ 2.9 และ สมการที่ 2.10 ในรูปเวกเตอร์ดังนี้

$$y_k = X_k^T W_k = W_k^T X_k \quad (3.6)$$

### 3.2 ผลตอบสนองที่ต้องการ และค่าความผิดพลาด

ในระบบโดยทั่ว ๆ ไป จะมีลักษณะการควบคุมสัญญาณอยู่สองลักษณะ  
ลักษณะแรกคือ การควบคุมวงเปิด ระบบจะไม่นำสัญญาณเอาท์พุทป้อนกลับมาเพื่อ  
การปรับสัญญาณ และอีกลักษณะหนึ่งคือ การควบคุมแบบวงปิด ระบบจะนำส่วนของ  
สัญญาณเอาท์พุทป้อนกลับมาควบคุมตัวเอง และสำหรับระบบปรับสถานะ จะต้องมีย่อ  
มูลอื่นอีก เพื่อการปรับสัญญาณให้เอาท์พุทเป็นตามต้องการ

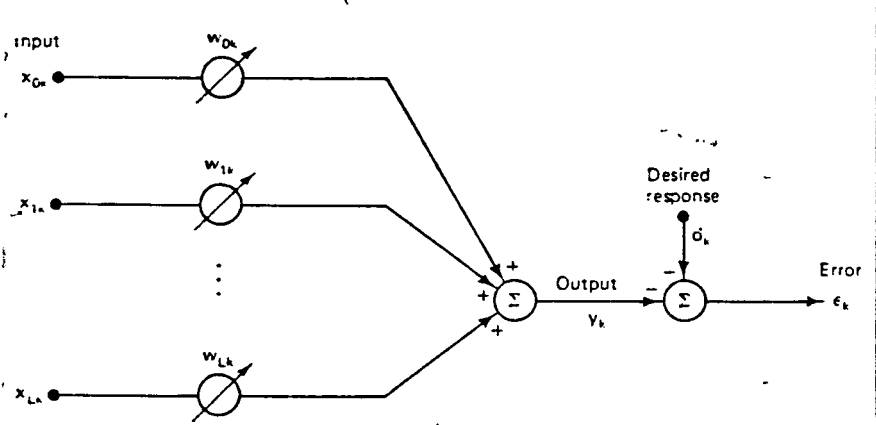


รูปที่ 3.3 การปรับสถานะแบบวงปิด



การที่ต้องมีข้อมูลอื่น ๆ พิเศษของระบบปรับสถานะ สัญญาณนี้ เรียกว่า สัญญาณผลตอบแทนที่ต้องการ (desired response) สัญญาณเทรนนิ่ง (training signal) ในระหว่างการปรับสถานะ ค่าถ่วงดุลของระบบปรับสถานะ จะถูกปรับเพื่อให้ค่าสัญญาณเอาต์พุต  $y_k$  มีค่าใกล้เคียงกับสัญญาณผลตอบแทนที่ต้องการมากที่สุด เพื่อให้ สัญญาณผลต่าง หรือ สัญญาณความผิดพลาด (ระหว่าง ค่าเอาต์พุตกับ ค่าที่ต้องการ) มีค่าน้อยที่สุดด้วยการปรับค่าสัมประสิทธิ์ของฟิลเตอร์ ซึ่งในการออกแบบนั้น จะให้การปรับสัมประสิทธิ์ ทำให้ค่าเฉลี่ยกำลังสอง (mean-square value) หรือค่ากำลังงานเฉลี่ย ของสัญญาณความผิดพลาดมีค่าน้อยที่สุด

วิธีการที่จะได้สัญญาณความผิดพลาด จากสัญญาณผลตอบแทนที่ต้องการ ได้แสดงในรูปที่ 3.4 ซึ่งเป็นกรณีของระบบหลายอินพุต ซึ่งแหล่งที่มาของผลตอบแทนที่ต้องการ ขึ้นอยู่กับลักษณะการใช้งานของวงจร ซึ่งจะได้อธิบายในบทต่อ ๆ ไป



รูปที่ 3.4 ระบบปรับสถานะ แสดงสัญญาณความผิดพลาด

3.3 ฟังก์ชันประเมินผล (Performance Function) และเส้นโค้งการเรียนรู้

ในหัวข้อนี้จะหาค่าความผิดพลาด โดยจะจัดรูปฟังก์ชันสัญญาณความผิดพลาด ให้เป็นฟังก์ชันของ ค่าถ่วงดุล

จากรูปที่ 3.4 สัญญาณความผิดพลาด ที่ดัชนีเวลา  $k$  ( $t=k$ ) แสดงได้ว่า

$$e_k = d_k - y_k \tag{3.7}$$

หลังจากแทนค่า  $y_k$  จากสมการที่ 3.6 ได้

$$e_k = d_k - X_k^T W = W^T X_k \tag{3.8}$$

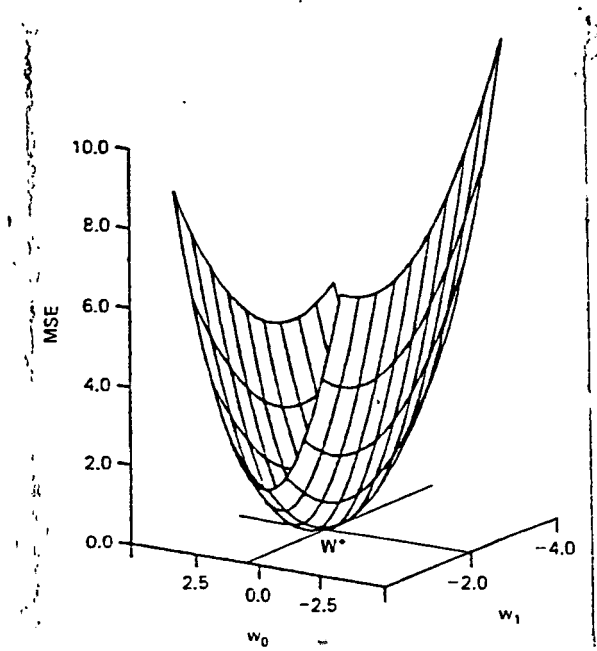
โดยที่  $k$  ของค่าถ่วงดุล ( $W$ ) ไว้ เนื่องจากจะพิจารณาขณะที่ยังไม่ปรับค่าถ่วงดุล เมื่อยกกำลังสองสมการที่ 3.8 ได้ค่าความผิดพลาดกำลังสองที่ขณะเวลานั้น ๆ

$$e_k^2 = d_k^2 + W^T X_k X_k^T W - 2d_k X_k^T W \quad (3.9)$$

โดยที่สมมติว่า  $e_k$ ,  $d_k$  และ  $X_k$  มีค่าคงที่ทางสถิติ (statistically stationary) และเมื่ออาศัยการใช้ค่าคาดหวัง (expected value) ทั้งสองข้างของสมการ ซึ่งค่าความผิดพลาดเฉลี่ยกำลังสอง (mean-square error) คือค่าคาดหวังของความผิดพลาดกำลังสอง

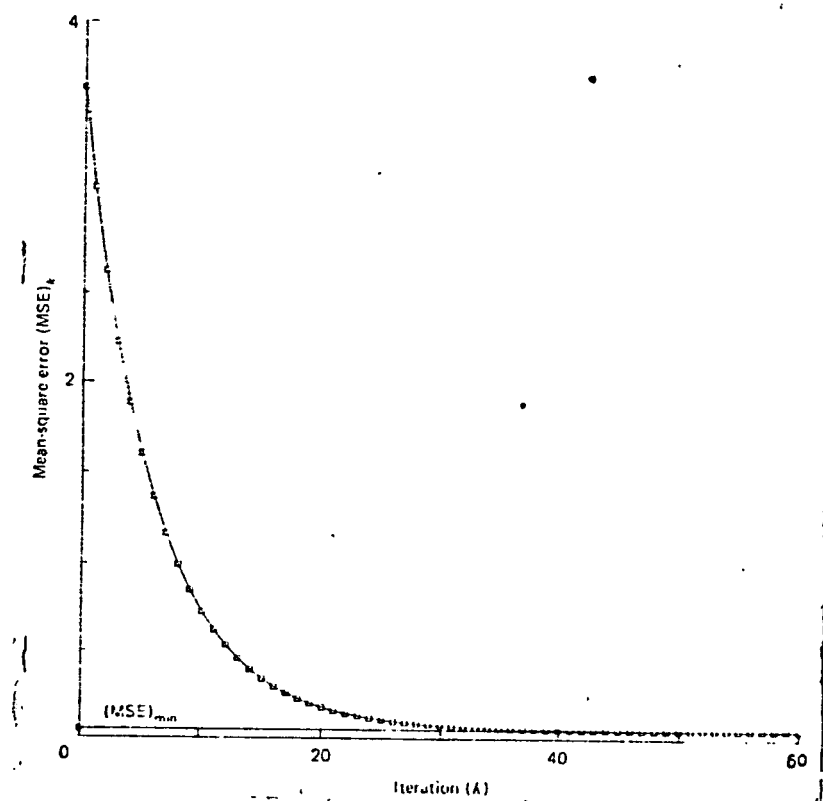
$$MSE \triangleq E[e_k^2] = E[d_k^2] + W^T E[X_k X_k^T] W - 2E[d_k X_k^T] W \quad (3.10)$$

ซึ่งในระบบปรับสถานะ เราต้องการปรับค่าถ่วงดุลเพื่อให้ค่าความผิดพลาดเฉลี่ยกำลังสองนี้น้อยที่สุด และสามารถเห็นได้จากสมการข้างต้นว่า ค่าความผิดพลาดเฉลี่ยกำลังสอง เป็นฟังก์ชันของค่าถ่วงดุลกำลังสอง ตัวอย่างของฟังก์ชันความผิดพลาดเฉลี่ยกำลังสอง แบบสองมิติ (ค่าถ่วงดุลมี 2 ฟังก์ชัน) แสดงในรูปที่ 3.5 ซึ่งเป็นรูป พาราโบลอยด์ หายชั้น ฟังก์ชันนี้เรียกอีกอย่างหนึ่งว่าฟังก์ชันประเมินผล



รูปที่ 3.5 แสดงตัวอย่างของ performance surface

ค่าความผิดพลาดเฉลี่ยกำลังสอง ในสมการที่ 3.10 ควรจะมีค่าลดลงเรื่อย ๆ ในขณะที่ระบบปรับสถานะทำงาน ซึ่งรูปกราฟของค่าความผิดพลาดเฉลี่ยกำลังสอง ที่เป็นฟังก์ชันของจำนวนการทำซ้ำ ( $k$ ) เรียกว่า เส้นโค้งการเรียนรู้ (learning curve) เนื่องจากเส้นโค้งนี้แสดงความสามารถของระบบในการตาม (track) สัญญาณผลตอบสนองที่ต้องการ ตัวอย่างของเส้นโค้งการเรียนรู้แสดงในรูปที่ 3.6



รูปที่ 3.6 แสดงตัวอย่างของ learning curve

### 3.4 Least-Mean-Square Algorithm

หัวข้อที่ผ่านมาได้กล่าวถึง ฟังก์ชันประเมินผล ซึ่งเป็นฟังก์ชันของค่าถ่วงดุล และได้กล่าวถึงการทำงานของระบบปรับสถานะ ที่ปรับทรานสเฟอร์ฟังก์ชันของตัวเอง โดยปรับสัมประสิทธิ์ของฟิลเตอร์ เพื่อให้การทำงานของระบบเข้าสู่จุดต่ำสุดของ ฟังก์ชันประเมินผล หัวข้อต่อไปนี้จะพิจารณาวิธีเข้าสู่จุดต่ำสุดของฟังก์ชันประเมินผล โดยวิธี ค่าเฉลี่ยกำลังสองต่ำสุด (Least-mean-square algorithm) หรือ LMS algorithm โดย Widrow [1] ได้ให้ความเห็นว่า วิธี LMS algorithm เป็นวิธีที่ดีที่สุด สำหรับระบบที่มีโครงสร้างเป็น วงจรรวมแบบเชิง

เส้น

หลักการของการเข้าสู่จุดต่ำสุดในฟังก์ชันประเมิณผล (performance function) ของวงจรวมเชิงเส้น หรือ ฟิลเตอร์ คือ ปรับค่าถ่วงคูลของฟิลเตอร์ เพื่อให้ค่าความผิดพลาดเฉลี่ย (ระหว่าง ผลตอบสนองที่ต้องการ กับ เอาท์พุท) กำลังสอง (mean-square error) มีค่าน้อยที่สุด โดยปรับค่าถ่วงคูลแต่ละขณะเวลา ตามทิศทางตรงข้ามกับ แกรเดียนท์ ของฟังก์ชันประเมิณผล เขียนเป็นสมการได้ดังนี้

$$W_{k+1} = W_k - \mu \nabla_k \quad (3.11)$$

เมื่อ  $W_k$  เป็นเวกเตอร์ของค่าถ่วงคูล (สัมประสิทธิ์) ของฟิลเตอร์ ที่เวลา  $t=k$

เวกเตอร์ของค่าถ่วงคูล :  $W_k = [w_0(k) \dots w_L(k)]^T$

แกรเดียนท์เวกเตอร์ :  $\nabla_k = \frac{\partial E[e_k^2]}{\partial W_k}$

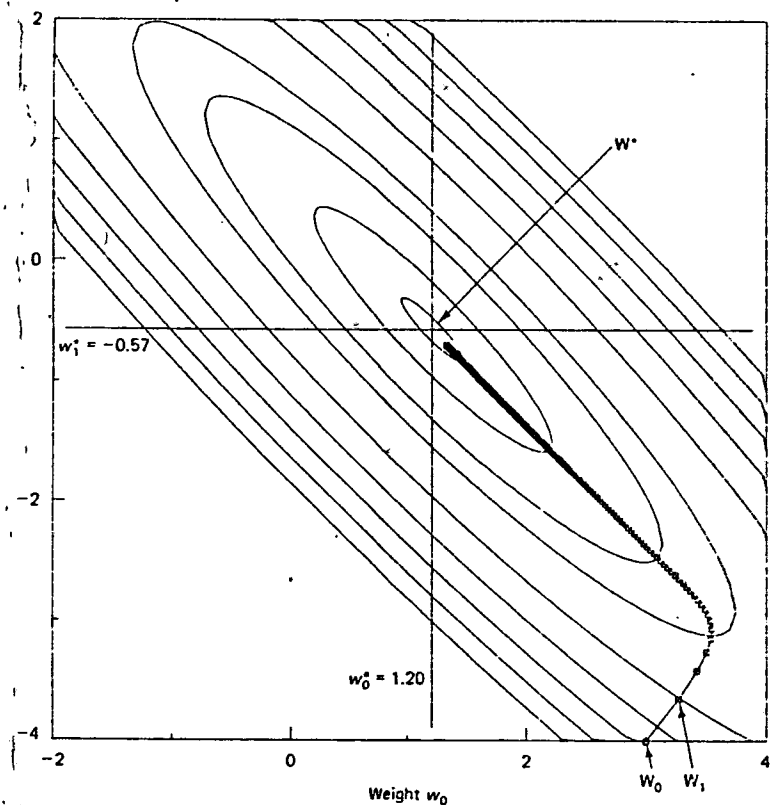
$$= \begin{bmatrix} \frac{\partial E[e_k^2]}{\partial w_0(k)} & \dots & \frac{\partial E[e_k^2]}{\partial w_L(k)} \end{bmatrix}^T$$

ค่า  $\mu$  เป็นพารามิเตอร์ ที่ควบคุมอัตราการลู่เข้าของระบบสู่จุดต่ำสุดของฟังก์ชันประเมิณผล

$e_k^2$  เป็นค่าความผิดพลาดกำลังสองที่เวลา  $t=k$

จากสมการที่ 3.11 เห็นได้ว่า ค่าถ่วงคูลจะปรับตัวเองเป็นสัดส่วนกับลบแกรเดียนท์ ( $-\nabla_k$ ) ของฟังก์ชันประเมิณผล ดังนั้นเมื่อ รู้ค่าแกรเดียนท์  $\nabla_k$  ที่ทุก ๆ  $k$  ในระหว่างการปรับค่าถ่วงคูล ก็จะทำให้ ค่าความผิดพลาดเฉลี่ยกำลังสอง มีค่าน้อยลงเรื่อย ๆ จนถึงจุดที่ แกรเดียนท์มีค่าเป็นศูนย์ คือ เมื่อ ค่าความผิดพลาดเฉลี่ยกำลังสอง มีค่าน้อยที่สุด ค่าถ่วงคูลก็จะมีค่าคงที่ เพื่อรักษาสภาพความผิดพลาดที่น้อยที่สุดนั้นไว้ ตัวอย่างของการเข้าสู่จุดต่ำสุดของฟังก์ชันประเมิณผล เมื่อรู้ค่าแกรเดียนท์ที่แน่นอน แสดงในรูปที่ 3.6

ในทางปฏิบัติจริง ๆ แล้ว วิธีการปรับค่าถ่วงคูลตามสมการที่ 3.11 จะทำได้ยาก หรือทำไม่ได้ เพราะไม่สามารถที่จะรู้ แกรเดียนท์ของฟังก์ชันประเมิณผลได้ล่วงหน้าได้ ดังนั้นจึงต้องใช้ในการประมาณค่าแกรเดียนท์ (gradient estimate) แทนในส่วน of ค่าแกรเดียนท์ในสมการ ที่ 3.11



รูปที่ 3.6 แสดงการลู่เข้าของระบบเมื่อรู้ค่าแกรเดียนต์แน่นอน

การประมาณค่าแกรเดียนต์จะใช้ ค่าผิดพลาดกำลังสองชั่วขณะ (instantaneous squared error) [2] มาประมาณค่า

$$\text{gradient estimate} : \hat{\nabla} = \frac{\partial e_k^2}{\partial W_k} \quad (3.12)$$

แทนค่า  $e_k$  จากสมการที่ 3.7 ลงในสมการ 4.2 ได้

$$\hat{\nabla} = 2e_k \frac{\partial (d_k - y_k)}{\partial W_k} \quad (3.13)$$

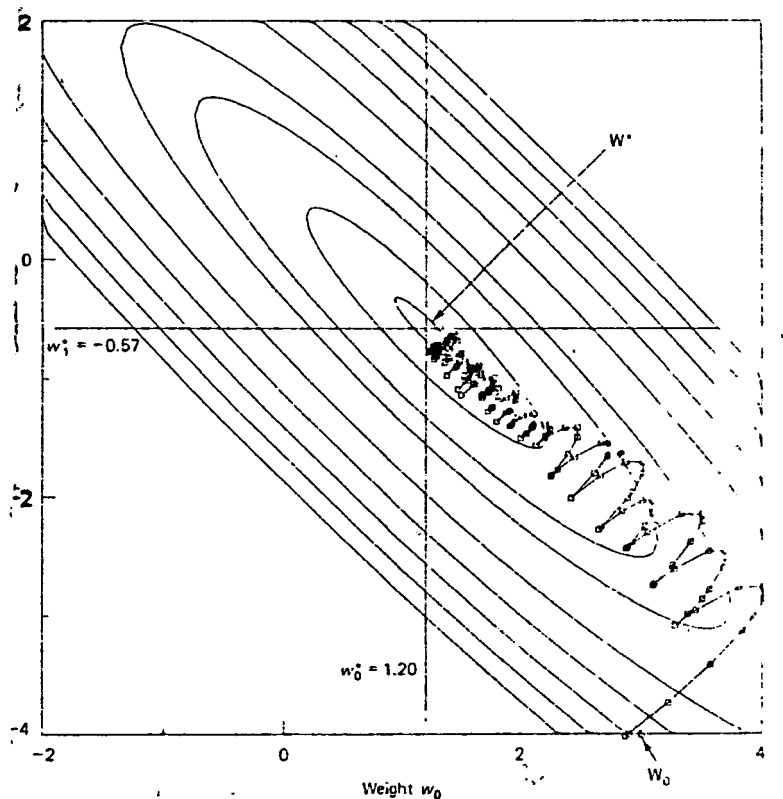
เนื่องจากผลตอบสนองที่ต้องการ ( $d_k$ ) ไม่เป็นฟังก์ชันของ ค่าถ่วงดุล และแทนค่า  $y_k$  จากสมการ 3.6 ได้

$$\hat{\nabla} = -2e_k X_k \quad (3.14)$$

เมื่อ  $X_k$  เป็นเวกเตอร์ของสัญญาณอินพุท และเมื่อแทนค่า  $\hat{\nabla}$  ลงในสมการที่ 4.1 แทน  $\nabla$  จะได้สมการที่ 4.5 ซึ่งเป็นสมการหลักของ LMS algorithm ของระบบปรับสถานะที่ได้ทำการศึกษา

$$W_{k+1} = W_k + 2\mu e_k X_k \quad (3.15)$$

ผลของการประมาณค่าแอมพลิจูด จะทำให้การทำงานของระบบปรับสถานะ optimize ได้โดยตรง. เพราะความผิดพลาดจากการประมาณค่าแอมพลิจูดนี้ รูปที่ 3.7 แสดงตัวอย่างของระบบปรับสถานะที่ใช้การประมาณค่าแอมพลิจูด ตามสมการที่ 3.12



รูปที่ 3.7 แสดงการลู่เข้าของระบบเมื่อใช้การประมาณค่าแอมพลิจูด

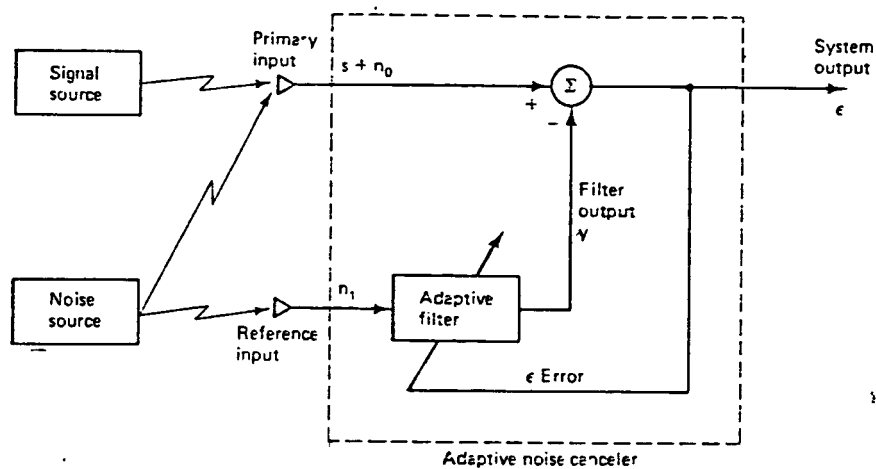
พิจารณาจากสมการที่ 3.15 พบว่าระบบปรับสถานะสามารถทำงานได้โดยที่มีสัญญาณอินพุต ( $X_n$ ) และสัญญาณผลตอบสนองที่ต้องการ ( $d_n$ ) เพื่อหาค่าความผิดพลาด ( $e_n$ ) ก็จะสามารถทำงานได้ เหลือเพียงแต่กำหนดค่าพารามิเตอร์ของการลู่เข้า  $\mu$  เท่านั้น

ค่าพารามิเตอร์ของการลู่เข้า  $\mu$  เป็นตัวสำคัญ ที่กำหนดความสามารถในการทำงานของระบบปรับสถานะ เนื่องจากการประมาณค่าแอมพลิจูดนั้น ได้มาจาก การใช้ค่าความผิดพลาดชั่วขณะกำลังสอง ( $e_n^2$ ) ไม่ใช่ จาก ค่าเฉลี่ยกำลังสอง ( $E[e_n^2]$ ) ถึงแม้ว่า โดยทั่วไปแล้วสมการที่ 3.15 สามารถทำให้ค่าถ่วงดุลสามารถเข้าสู่จุดทำงานที่ต้องการได้ แต่การเปลี่ยนค่าถ่วงดุลแต่ละครั้งก็ทำ

## บทที่ 4

### การประยุกต์ใช้ระบบปรับสถานะในการกำจัดสัญญาณรบกวน

วงจรกำจัดสัญญาณรบกวนปรับสถานะแบบเบื้องต้น ได้แสดงในรูปที่ 4.1 ซึ่งอุปกรณ์รับสัญญาณ (input sensors) จะเป็นชนิดใดก็ได้ที่เป็นชนิดเดียวกันทุกอัน เช่น ไมโครโฟน หรือ สายอากาศ เป็นต้น



รูปที่ 4.1 หลักการของระบบกำจัดสัญญาณรบกวนปรับสถานะ

ตามรูปที่ 4.1 หลักการของระบบกำจัดเสียงรบกวนปรับสถานะ มีดังนี้ สัญญาณที่ต้องการ (signal) ถูกส่งผ่านช่องทางสื่อสารใด ๆ เช่น เป็นเสียงผ่าน มาในสายโทรศัพท์ เข้ามาทางอุปกรณ์รับ รับสัญญาณที่ต้องการ ( $s$ ) ที่ถูกรบกวน โดยสัญญาณรบกวน ( $n_0$ ) ที่มีไม่ความสัมพันธ์กัน กับสัญญาณ  $s$  (signal  $s$  uncorrelated with noise  $n_0$ ) โดยสัญญาณนี้ ( $s+n_0$ ) เข้ามาทางอินพุตหลัก (primary input) ของวงจรกำจัดสัญญาณรบกวน และ อุปกรณ์รับตัวที่สอง จะรับแต่เฉพาะสัญญาณรบกวน ( $n_1$ ) โดยที่  $n_1$  ไม่สัมพันธ์กัน (uncorrelated) กับ สัญญาณที่ต้องการ แต่มีความสัมพันธ์กัน (correlated) บ้างกับสัญญาณ  $n_0$  เช่น เป็นสัญญาณรบกวนที่มาจากแหล่ง เดียวกัน หรือ จากแหล่งที่มีลักษณะคล้ายกัน โดยที่สัญญาณ  $n_1$  นี้จะเข้ามา ทางอินพุตอ้างอิง (reference input) ของวงจร สัญญาณรบกวน  $n_1$  จะ ถูกกรอง (filter) โดยวงจรปรับสถานะจะพยายามทำให้เอาท์พุท  $y$  มี คุณสมบัติใกล้เคียงกับ  $n_0$  จากนั้นสัญญาณ  $y$  นี้ จะถูกนำไปลบออกจาก สัญญาณอินพุต

หลัก  $(s+n_0)$  ได้เป็นเอาต์พุทของระบบ เท่ากับ  $s+n_0-y$

ถ้าสามารถทราบได้ล่วงหน้าถึง คุณลักษณะของสัญญาณรบกวน และ คุณลักษณะของช่องสัญญาณ (channel) ที่สัญญาณรบกวนถูกส่ง (transmit) มาที่ อุปกรณ์รับอินพุทหลัก และ อุปกรณ์รับอินพุทอ้างอิง ก็จะสามารถที่จะออกแบบวงจรดิจิทัลฟิลเตอร์แบบธรรมดา ที่ค่าถ่วงดุลเป็นค่าคงที่ เพื่อทำให้ฟิลเตอร์ มีค่าเอาต์พุท เป็น  $y=n_0$  และเมื่อนำเอาต์พุท  $y$  ไปลบออกจากสัญญาณอินพุทหลัก  $(s+n_0)$  จะเหลือเพียงสัญญาณที่ต้องการ  $(s)$  ออกที่เอาต์พุทของระบบกำจัดสัญญาณรบกวน

แต่เนื่องจากคุณลักษณะของช่องสัญญาณ ที่สัญญาณรบกวนเดินทางผ่านมานั้น ไม่สามารถทราบได้ล่วงหน้า หรือ คุณลักษณะของช่องสัญญาณนี้เปลี่ยนแปลงตามเวลา การใช้ดิจิทัลฟิลเตอร์ที่มีค่าถ่วงดุลคงที่ ก็จะเป็นการไม่เหมาะสม หรือว่าถึงแม้ว่าจะสามารถออกแบบ วงจรดิจิทัลฟิลเตอร์ได้ก็ตาม แต่เราก็จะยังยากที่จะปรับวงจร ให้ถูกต้องตามการคำนวณทุกประการ แต่วงจรที่เป็นระบบปรับสถานะ จะตัดความยุ่งยากเหล่านี้ออกไปได้ เนื่องจากระบบจากปรับตัวเองเข้าสู่จุดทำงานที่ดีที่สุดด้วยตัวเอง

เห็นได้ว่าสัญญาณต่าง ๆ ที่ใช้ในวงจรกำจัดสัญญาณรบกวนนี้ สัญญาณผลตอบสนองที่ต้องการ คือ สัญญาณอินพุทอ้างอิง ส่วนสัญญาณความผิดพลาด เป็นสัญญาณที่ใช้เป็นเอาต์พุทของระบบ ซึ่งอาจจะดูสับสน แต่ถ้าพิจารณาให้ดีจะพบว่า ฟิลเตอร์นี้จะพยายามทำให้สัญญาณรบกวน ตัดกันออกให้มากที่สุด โดยทำให้สัญญาณรบกวนที่ผ่าน ฟิลเตอร์มีความคล้ายคลึงกันกับสัญญาณรบกวน ที่ผสมมาในสัญญาณที่ต้องการมากที่สุด ส่วนเหลือจากการลบกันระหว่างสองสัญญาณนี้ จะได้แต่เฉพาะสัญญาณที่ต้องการ ดังนั้นจึงใช้สัญญาณความผิดพลาดมาเป็น เอาต์พุทของระบบ การใช้หลักการปรับสถานะนี้ มีข้อดีเหนือการใช้ดิจิทัลฟิลเตอร์ แบบค่าถ่วงดุลคงที่ มากำจัดสัญญาณรบกวน คือ ระบบปรับสถานะไม่จำเป็นต้องรู้คุณลักษณะของสัญญาณใด ๆ ล่วงหน้าเลย หรือไม่จำเป็นต้องรู้ความสัมพันธ์ (interrelationship) ระหว่างสัญญาณใด ๆ ล่วงหน้าเลยเช่นกัน

จากรูปที่ 4.1 สมมติให้สัญญาณ  $s, n_0, n_1$  และสัญญาณ  $y$  มีคุณลักษณะทางสถิติคงที่ (statistically stationary) และ ต่างก็มีค่าเฉลี่ยเป็นศูนย์ (zero mean) โดยที่สมมติว่า สัญญาณ  $(s)$  ไม่มีความสัมพันธ์ (uncorrelated) กับ  $n_0$  และ  $n_1$  และสมมติว่า  $n_1$  มีความสัมพันธ์ (coorelated) กับสัญญาณ  $n_0$  จะได้เอาต์พุท จากฟิลเตอร์

$$e = s + n_o - y \quad (4.1)$$

ยกกำลังสองสมการที่ 4.1 ได้

$$e^2 = s^2 + (n_o - y)^2 + 2s(n_o - y) \quad (4.2)$$

หาค่าคาดหวังทั้งสองข้างของสมการ 4.2 ได้

$$E[e^2] = E[s^2] + E[(n_o - y)^2] + 2E[s(n_o - y)] \quad (4.3)$$

เนื่องจาก  $s$  ไม่มีความสัมพันธ์กับ  $n_o, n_i$  และ  $y$  โดยที่ทุกสัญญาณมีค่าเฉลี่ยเป็นศูนย์

$$E[e^2] = E[s^2] + E[(n_o - y)^2] \quad (4.4)$$

ค่ากำลังงานของสัญญาณที่ต้องการ  $E[s^2]$  จะไม่เปลี่ยนแปลงจากการที่ฟิลเตอร์ทำการปรับค่าถ่วงดุล เพื่อที่จะลด ค่ากำลังงานของสัญญาณความผิดพลาด  $E[e^2]$  ดังนั้น กำลังงานของสัญญาณเอาต์พุต  $E[e^2]$  ที่ต่ำที่สุด คือ

$$E_{\min}[e^2] = E[s^2] + E_{\min}[(n_o - y)^2] \quad (4.5)$$

ดังนั้น เมื่อฟิลเตอร์ทำการปรับค่าถ่วงดุล เพื่อให้  $E[e^2]$  มีค่าต่ำที่สุด จะทำให้เฉพาะ  $E[(n_o - y)^2]$  มีค่าต่ำที่สุด คือระบบอยู่ในสภาวะ optimized หมายความว่า เอาต์พุตของฟิลเตอร์ ( $y$ ) เป็นค่าประมาณค่ากำลังสองต่ำสุด (least-squared estimate) ที่ถูกต้องที่สุดของสัญญาณรบกวนที่ผสมอยู่สัญญาณอินพุตหลัก และจากการย้ายข้างสมการที่ 4.1 ได้

$$e - s = n_o - y \quad (4.6)$$

เมื่อยกกำลังสองและหาค่าคาดหวังทั้งสองข้างของสมการ 4.6 ได้

$$E[(e - s)^2] = E[(n_o - y)^2] \quad (4.7)$$

การทำให้  $E[(n_o - y)^2]$  มีค่าต่ำสุดจะทำให้  $E[(e - s)^2]$  มีค่าต่ำสุดด้วยแสดงว่า การทำให้ระบบเข้าสู่ จุด optimize เป็นการทำให้สัญญาณความผิดพลาด  $e$  เป็นค่าประมาณค่ากำลังสองที่ต่ำที่สุด ของสัญญาณ  $s$  ที่ต้องการ สำหรับวงจรตามรูป 4.1

จากสมการที่ 4.1 สัญญาณความผิดพลาด  $e$  เป็นเอาต์พุตของวงจรกำจัดสัญญาณรบกวน โดยทั่วไปเอาต์พุตนี้จะมีสัญญาณ  $s$  ผสมกับสัญญาณรบกวน  $(n_o - y)$  อัตราส่วนของสัญญาณ ต่อ สัญญาณรบกวน (signal-to-noise ratio) ที่เอาต์พุตของระบบกำจัดสัญญาณรบกวน แสดงดังสมการ 4.8

$$S/N = E[s^2] / E[(n_o - y)^2] \quad (4.8)$$

กำลังงานของสัญญาณรบกวนที่เอาต์พุต  $E[(n_o - y)^2]$  จะถูกลดลงขณะที่ระบบลดกำลังงานของเอาต์พุต  $E[e^2]$  แต่กำลังงานของสัญญาณที่ต้องการ  $E[s^2]$

ไม่ได้ถูกทำให้เปลี่ยนแปลงจากการทำงานของกระบวนการนี้ ดังนั้นการทำงานของระบบกำจัดสัญญาณรบกวนปรับสถานะ จึงเป็นการเพิ่มอัตราส่วนของ สัญญาณต่อสัญญาณรบกวน (signal-to-noise ratio) ที่เอากำพุดของระบบ

## บทที่ 5

### ผลการจำลอง (Simulated results)

จากวงจรกำจัดสัญญาณรบกวนปรับสถานะในรูปที่ 4.1 โดยใช้ LMS algorithm จากสมการที่ 3.15 และสร้างสัญญาณต่าง ๆ ด้วยฟังก์ชันคณิตศาสตร์ โดยมีขั้นตอนดังนี้

1. สร้างสัญญาณที่ต้องการ ( $s$ )
2. ผสมสัญญาณรบกวน ( $n_0$ ) เข้ากับสัญญาณตามข้อ 1  
จะได้เป็น สัญญาณที่เข้ามาที่อินพุตหลัก ของวงจรตามรูปที่ 4.1 จากนั้นขั้นตอนต่อไปคือ
3. สร้างสัญญาณรบกวน ( $n_1$ ) ที่มีลักษณะคล้ายสัญญาณรบกวน ตามข้อ 2 เป็นสัญญาณที่เข้ามาที่อินพุตอ้างอิงของวงจร
4. นำสัญญาณอินพุตอ้างอิง และสัญญาณอินพุตหลัก มาทำการคำนวณด้วย LMS algorithm ได้ผลลัพธ์ของ ฟิลเตอร์ปรับสถานะ ( $y$ )
5. นำสัญญาณ  $y$  ไปลบออกจากสัญญาณ อินพุตหลัก ได้เป็นสัญญาณเอาต์พุตของระบบ
6. หาค่า ความผิดพลาดเฉลี่ยกำลังสอง (Mean-Square Error) ของสัญญาณเอาต์พุตจากวงจรกำจัดสัญญาณรบกวน โดยหาค่าเฉลี่ยผลต่างกำลังสอง  $E[(e-s)^2]$  ระหว่าง สัญญาณเอาต์พุตของวงจร กับ สัญญาณที่ต้องการ และจากสมการ ที่ 4.7 ค่าที่ได้นี้จะ เป็นค่ากำลังงานของสัญญาณรบกวน ที่เอาต์พุตของระบบ
7. หาค่า อัตราส่วนสัญญาณต่อสัญญาณรบกวน ของเอาต์พุตของระบบกำจัดสัญญาณรบกวนได้จาก การหาค่ากำลังงานของ สัญญาณ  $s$  แล้วนำค่ากำลังงานของสัญญาณรบกวนที่เอาต์พุต จากข้อ 6 มาหาร ตามสมการ

$$S/N \text{ ratio} = E[s^2]/E[(e-s)^2]$$

8. จุดที่ตัดสินการลู่เข้า คือเมื่อ Mean-square error น้อยกว่า 1% ของ ค่า error ที่มากที่สุด

การจำลองการทำงานของ ระบบกำจัดสัญญาณรบกวนได้แบ่งการทดลองออกเป็นตอน ๆ เพื่อ ศึกษาผลกระทบจาก การแปรค่าเฉพาะค่าใดค่าหนึ่งซึ่งรายละเอียดจะอยู่ในการทดลองตอนต่าง ๆ

## 5.1 สัญญาณที่ต้องการ (s) เป็นสัญญาณความถี่เดียว

ลักษณะของสัญญาณที่ต้องการ แสดงในรูปที่ 5.1 และตัวอย่างของสัญญาณ s ผสมสัญญาณรบกวน โดยมีค่า S/N เท่ากับ -4.495 dB แสดงในรูปที่ 5.2

### 5.1.1 การทดลองที่ 1 : ผลจากการแปรค่าพารามิเตอร์ของการลู่เข้า ( $\mu$ )

เงื่อนไขคงที่ทุกกรณี :

$$\text{สัญญาณอินพุตหลัก} = s + n_0 \quad \text{โดยที่ } s/n_0 = 1$$

$$\text{สัญญาณอินพุตอ้างอิง} = n_1 \quad \text{โดยที่ } n_1 = n_0$$

ตัวแปร : ค่าพารามิเตอร์ของการลู่เข้า  $\mu$

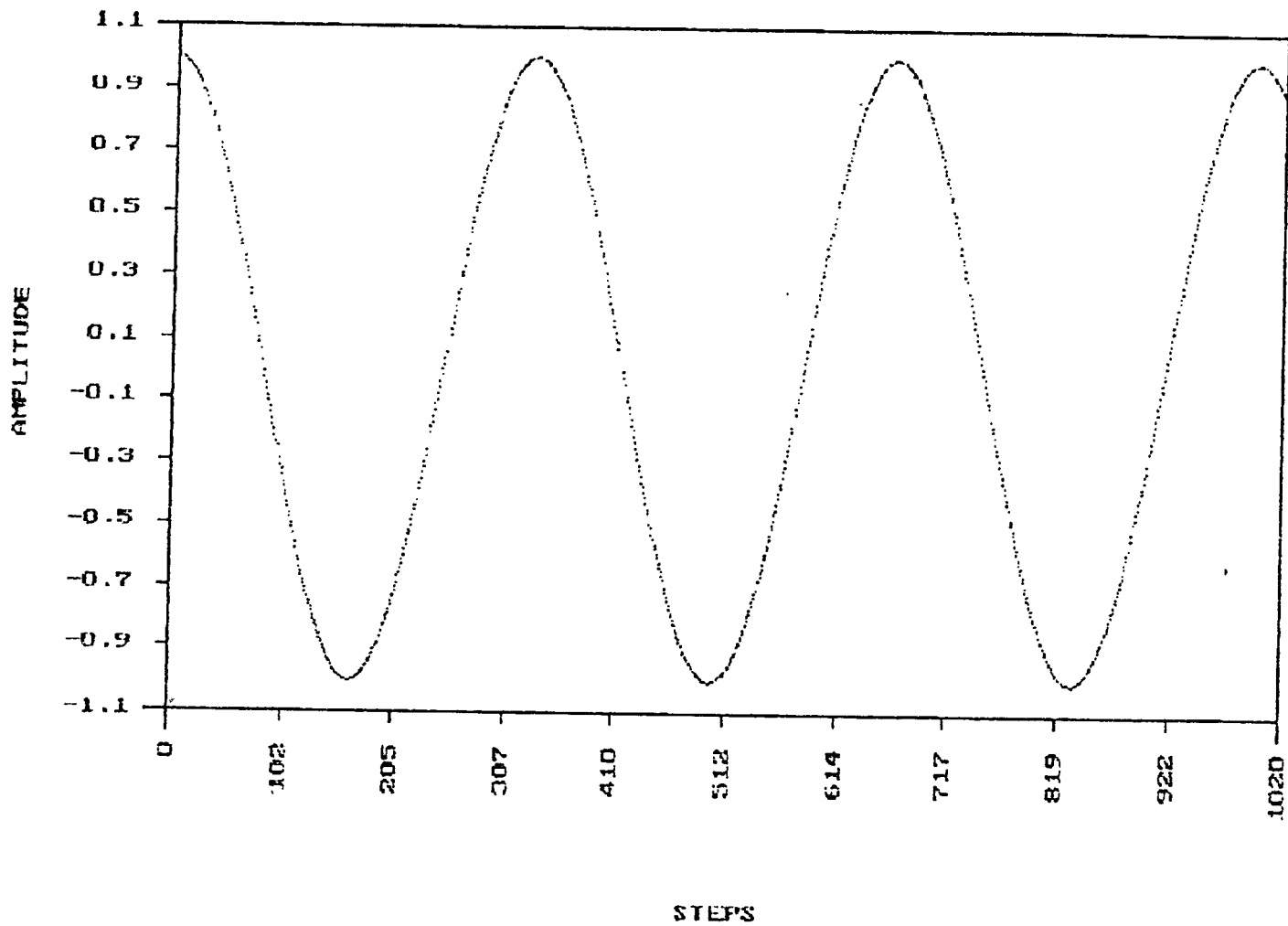
ได้ผลลัพธ์แสดงใน รูปที่ 5.3 ถึงรูปที่ 5.8 นำข้อมูลมาวิเคราะห์ แสดงในตารางที่ 5.1 ซึ่ง steps ในตาราง คือ จำนวนครั้งของการทำงานที่ค่า mean-square error ของสัญญาณเอาท์พุท น้อยกว่า 1% ของค่า error ที่มากที่สุด

$\mu$	steps	S/N (dB)
.002	500	-5.664
.004	300	8.219
.008	200	10.385

ตารางที่ 5.1

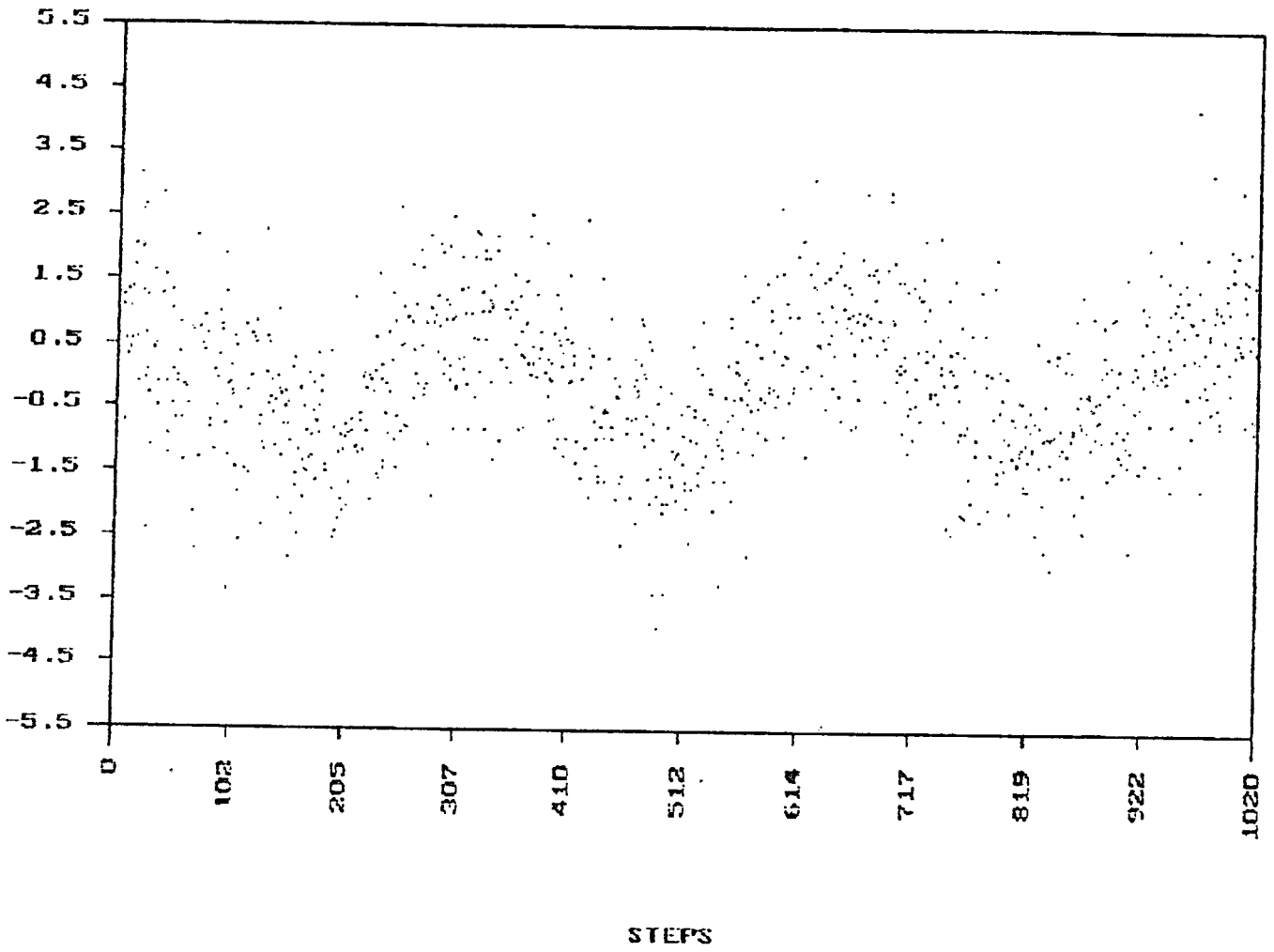
ข้อสังเกต เมื่อเพิ่มค่า  $\mu$  คือ ที่จุดยอด (peak) ของสัญญาณจะมีค่า ความผิดพลาดมาก แต่ระบบจะลู่เข้าเร็วกว่า

## SIGNAL



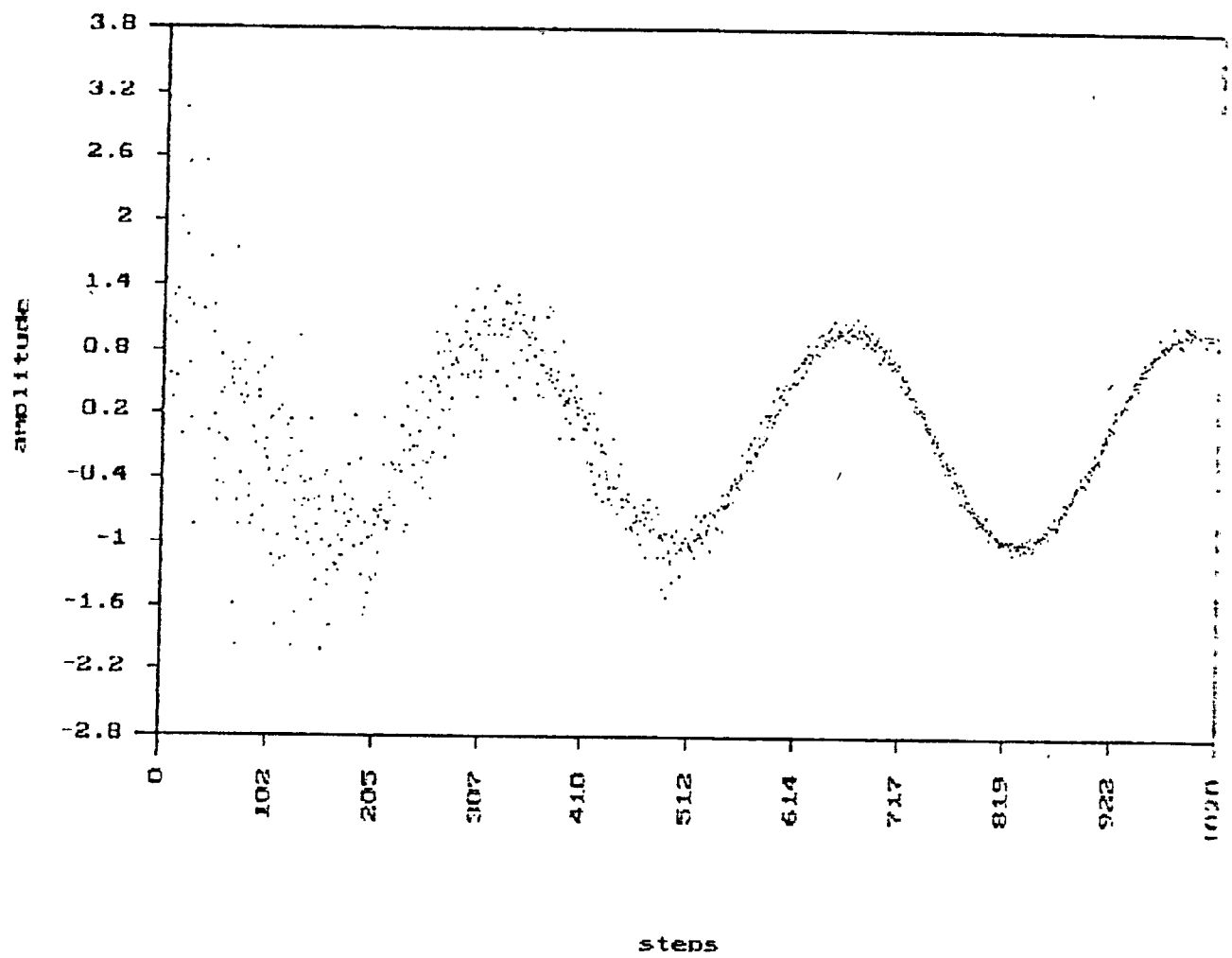
รูปที่ 5.1 สัญญาณที่ต้องการ (s)

SIGNAL + NOISE



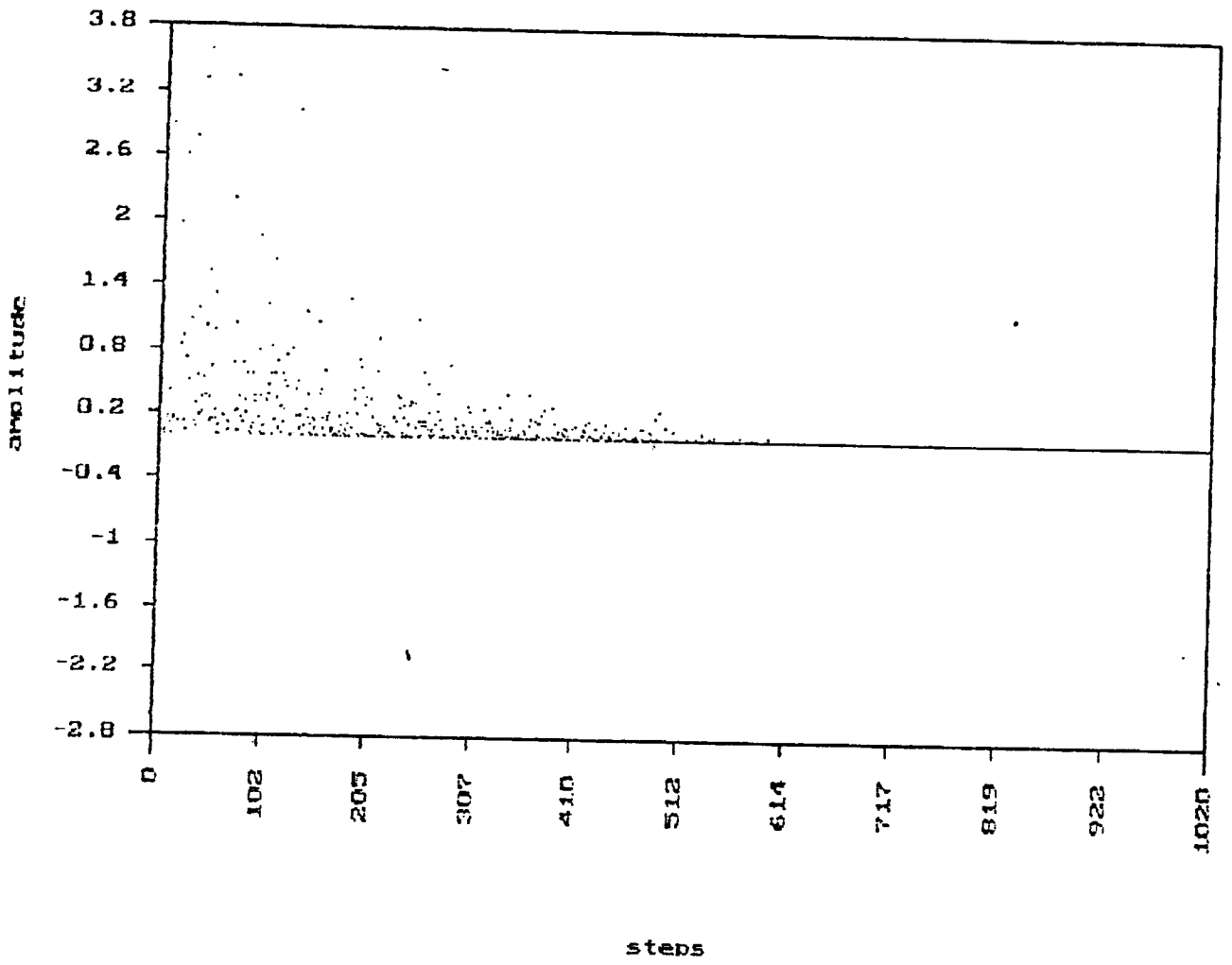
รูปที่ 5.2 สัญญาณอินพุทหลัก

$\mu=0.002$

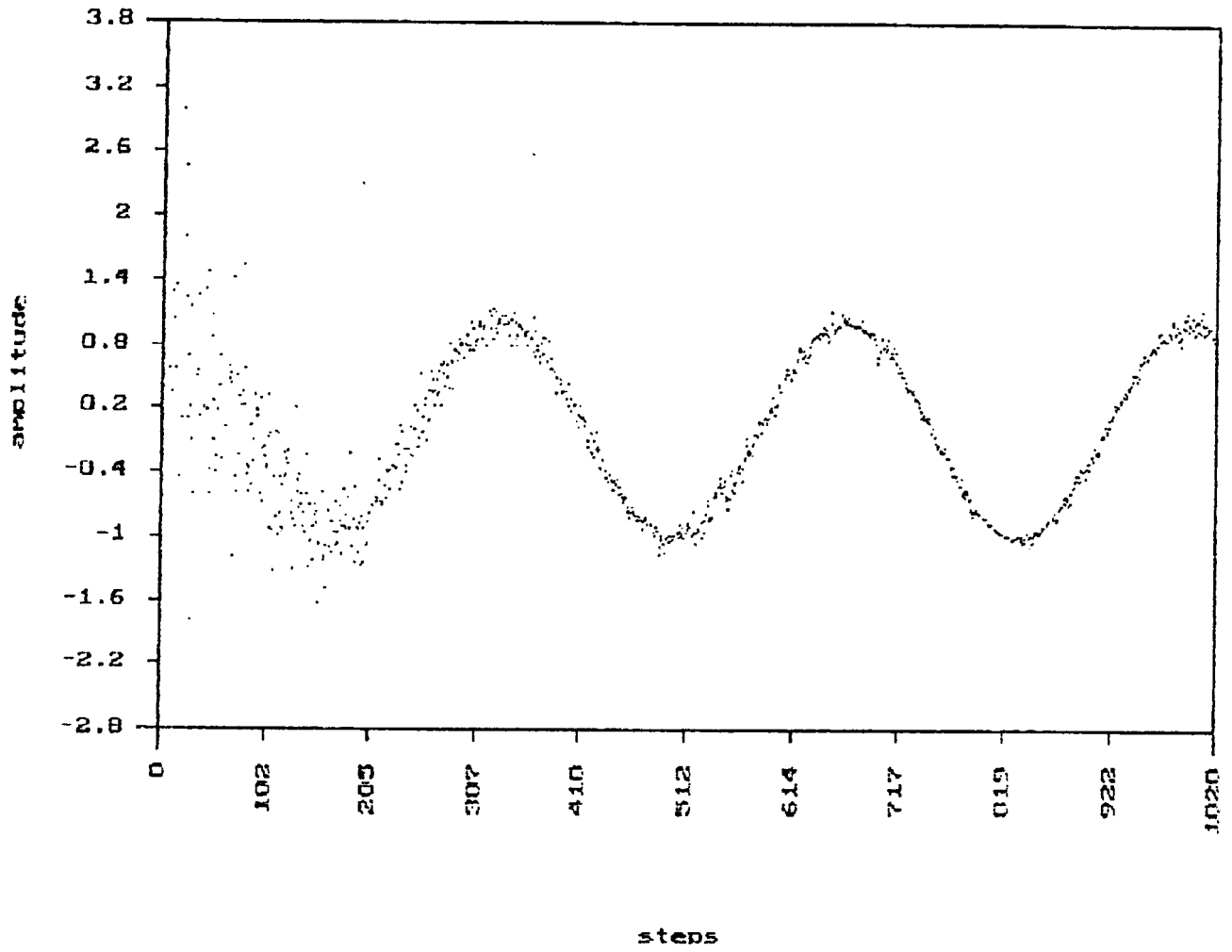


รูปที่ 5.3 สัญญาณเออร์เกทจากวงจรถ่ายสัญญาณรบกวน  $\mu = 0.002$

MEAN SQUARE ERROR ( $\mu=0.002$ )

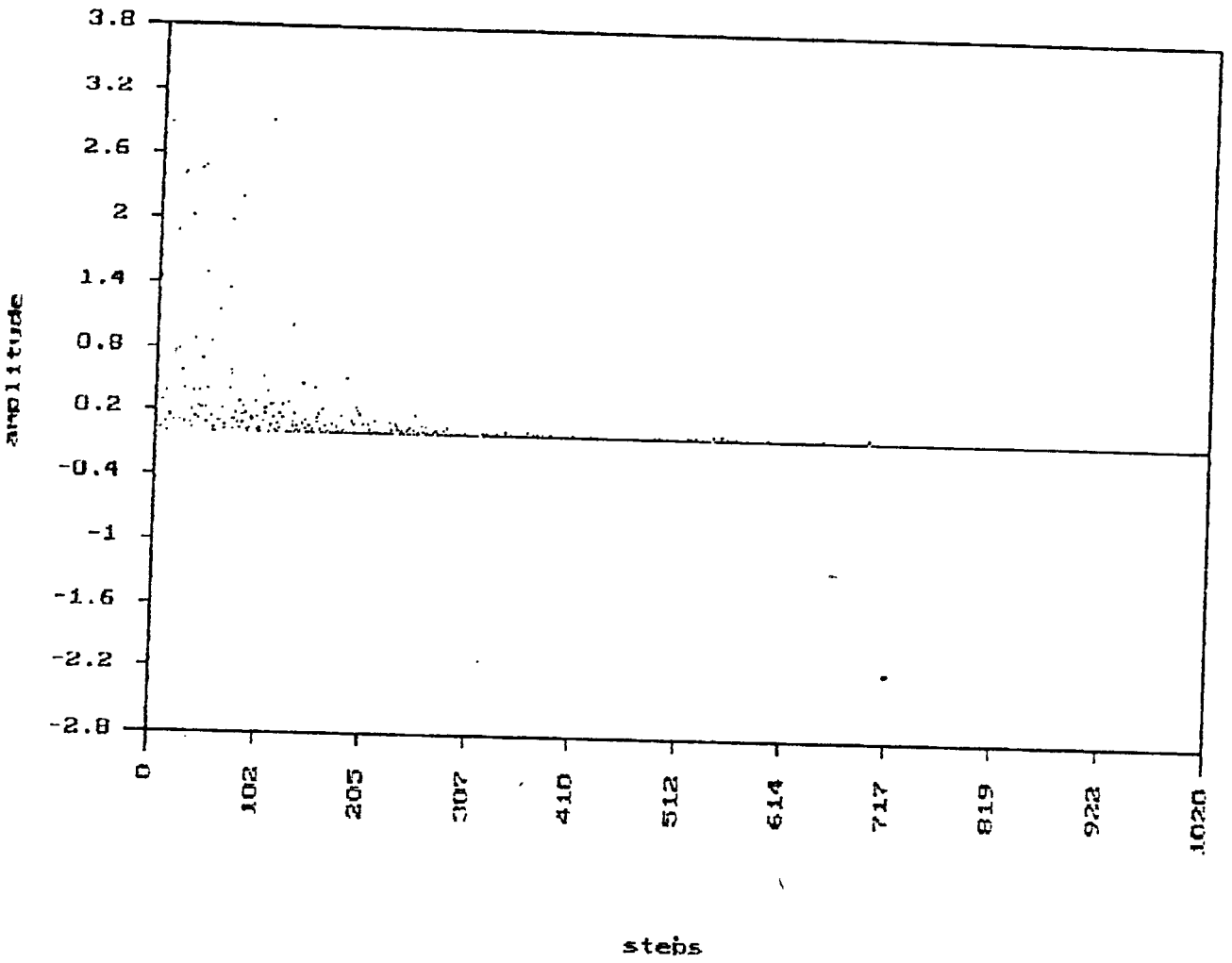


รูปที่ 5.4 กำลังงานของสัญญาณรบกวน  $\mu = 0.002$

$\mu = 0.004$ 

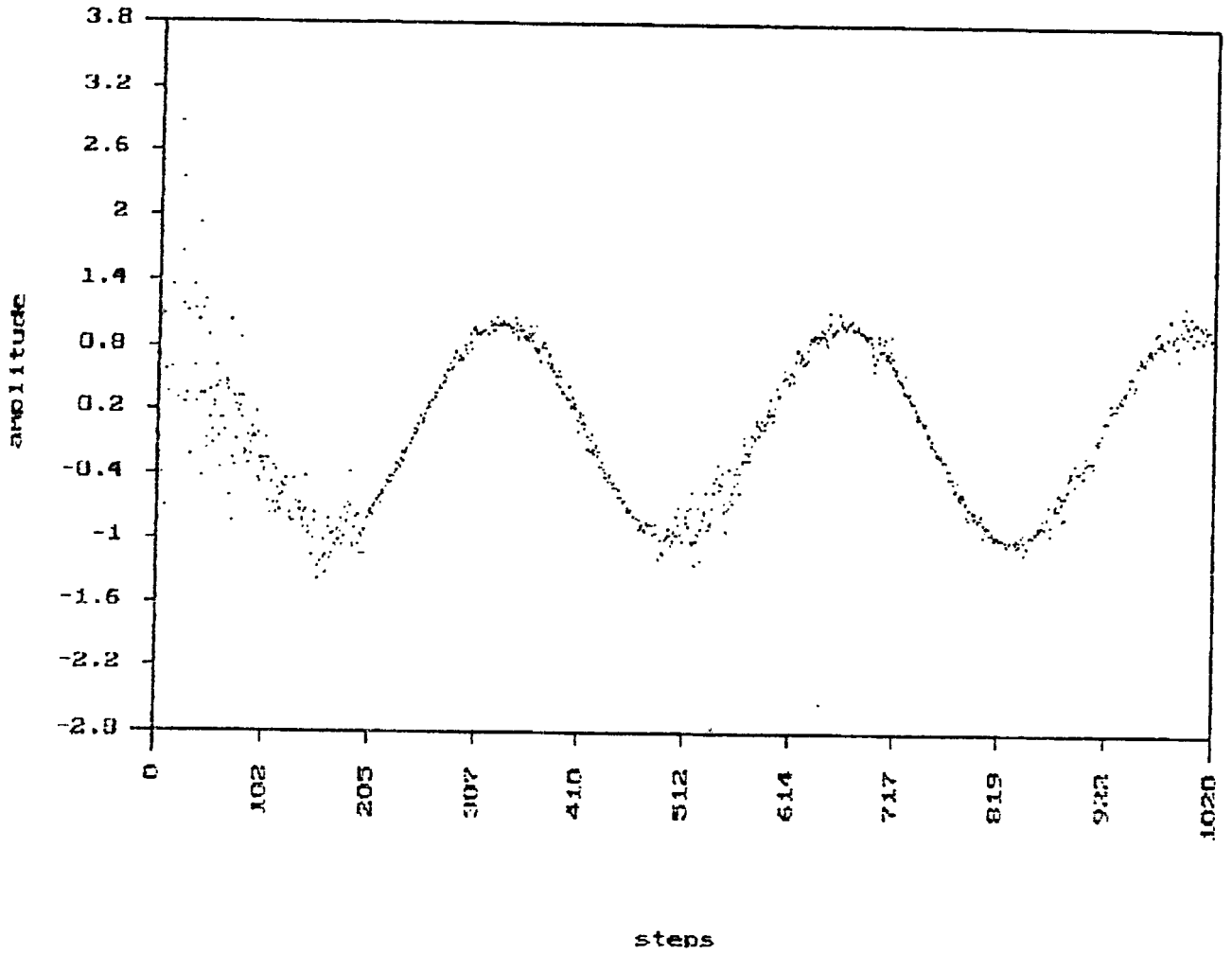
รูปที่ 5.5 สัญญาณเออร์กพัทจากวงจรกำจัดสัญญาณรบกวน  $\mu = 0.004$

MEAN SQUARE ERROR ( $\mu=0.004$ )

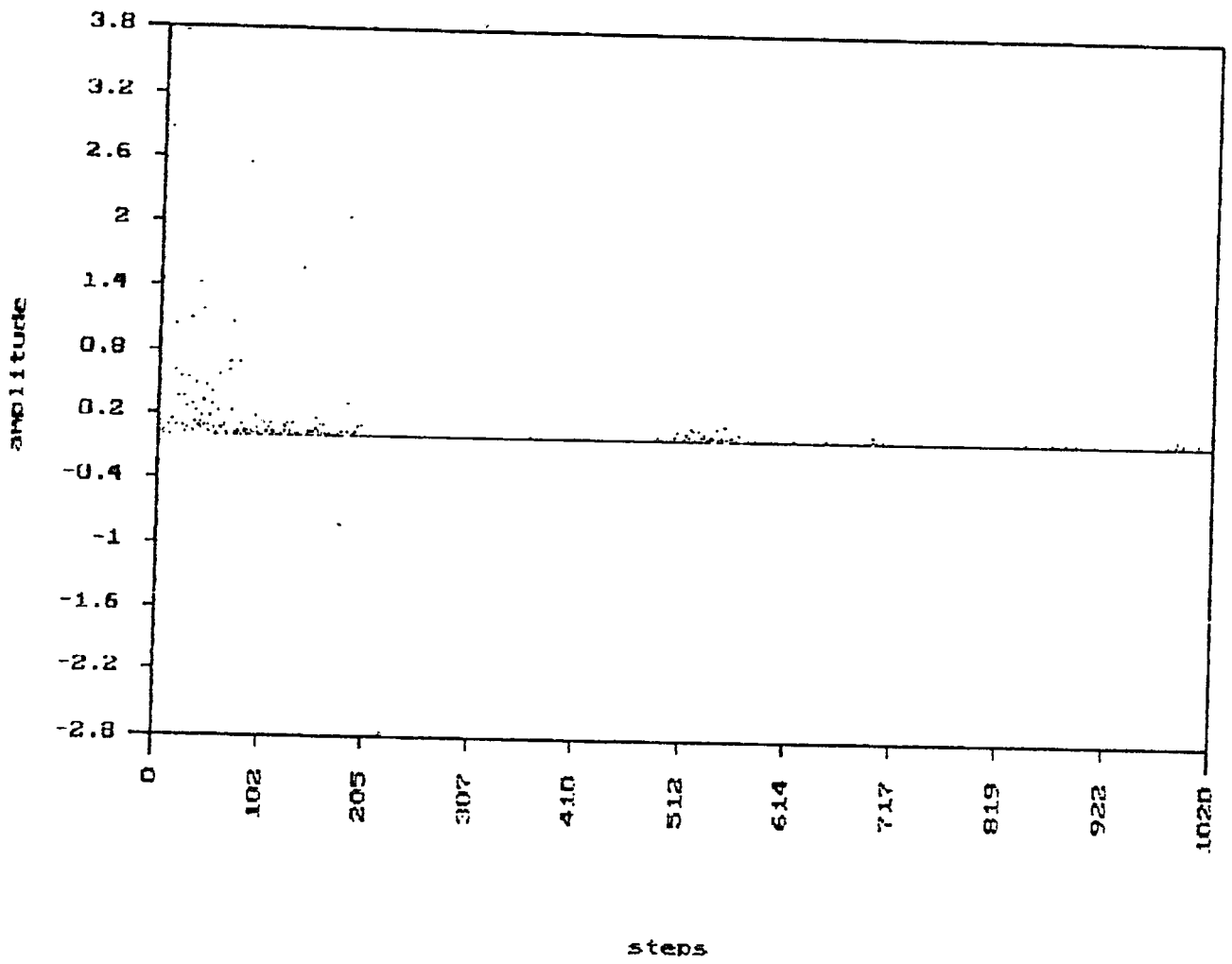


รูปที่ 5.6 กำลังงานของสัญญาณรบกวน  $\mu = 0.004$

$\mu = 0.008$



รูปที่ 5.7 สัญญาณเออร์เกนซ์ที่พ่วงจากวงจรกำเนิดสัญญาณรบกวน  $\mu = 0.008$



รูปที่ 5.8 กำลังงานของสัญญาณรบกวน  $\mu = 0.008$

### 5.1.2 การทดลองที่ 2 : ผลจากการแปรค่า ขนาดของอินพุทอ้างอิง

เงื่อนไขคงที่ทุกกรณี :

$$\text{สัญญาณอินพุทหลัก} = s+n_o \quad \text{โดยที่ } s/n_o = 1$$

$$\text{พารามิเตอร์ของการลู่เข้า } \mu = 0.001$$

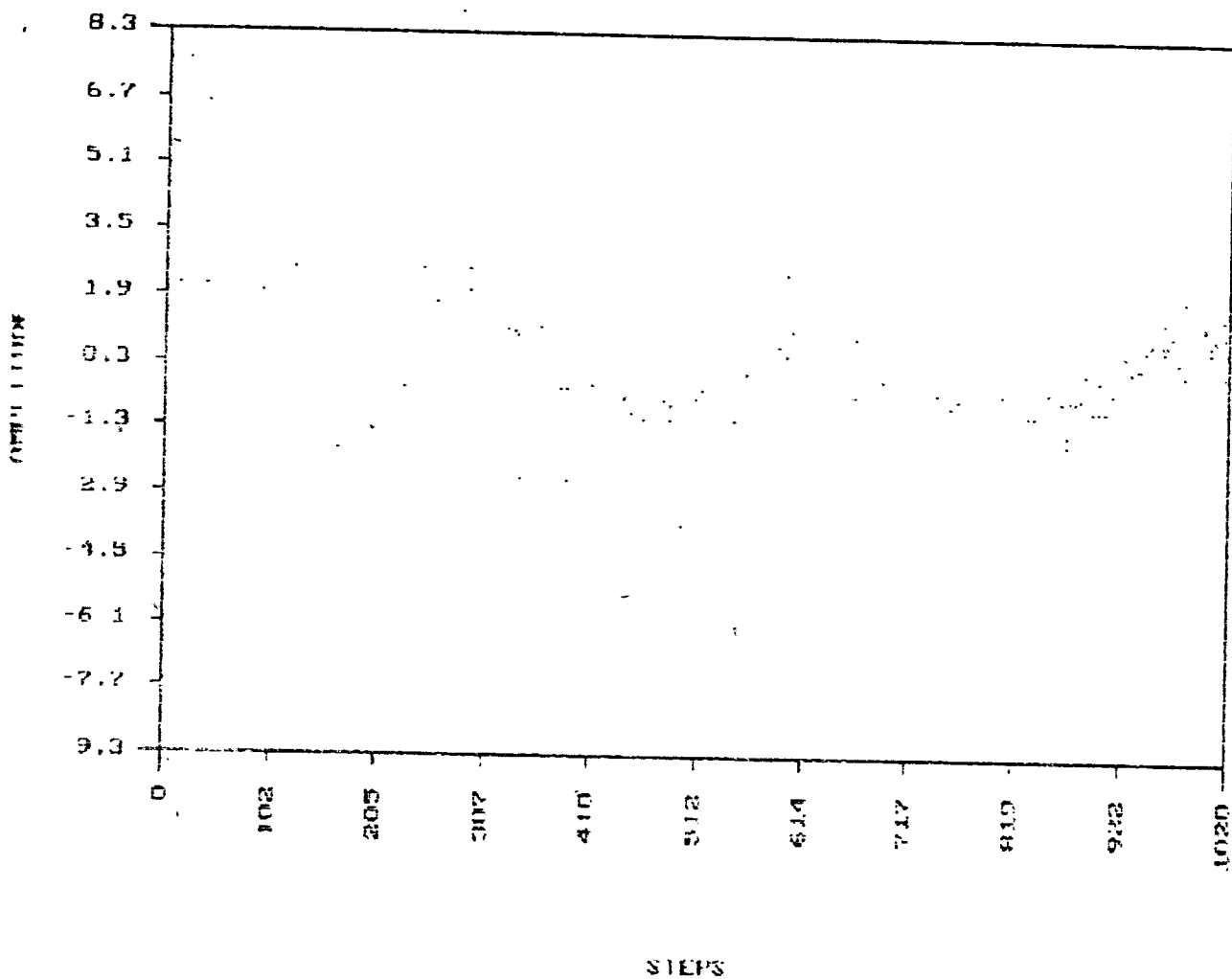
ตัวแปร : ขนาดของอินพุทอ้างอิง  $n_1$

ได้ผลลัพธ์แสดงในรูปที่ 5.9 ถึงรูปที่ 5.14 และสรุปผลแสดงในตารางที่ 5.2

$n_1/n_o$	steps	S/N (dB)
1	>1000	-6.360
2	250	0.676
3	200	2.562

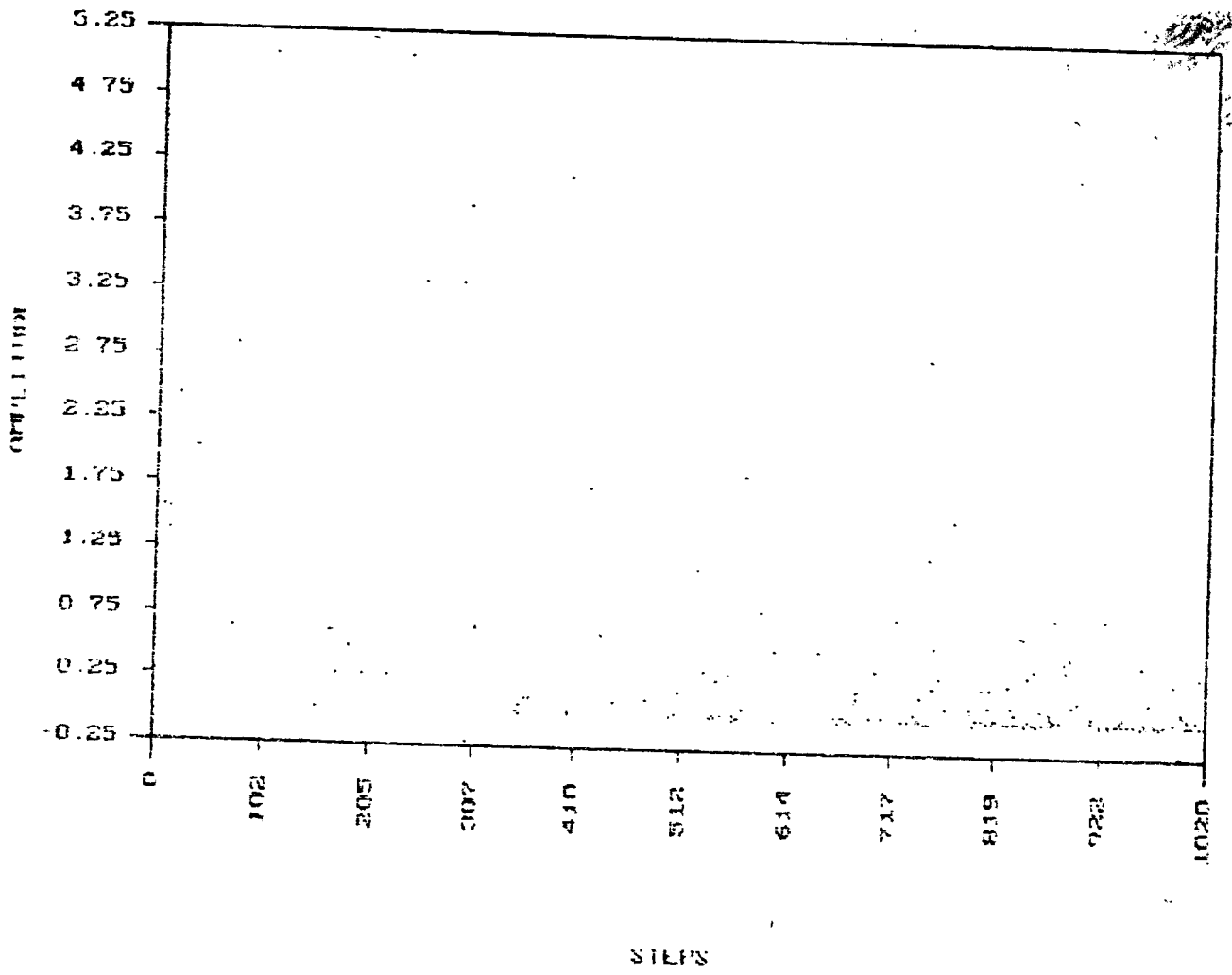
ตารางที่ 5.2

REFERENCE INPUT (N1)=1



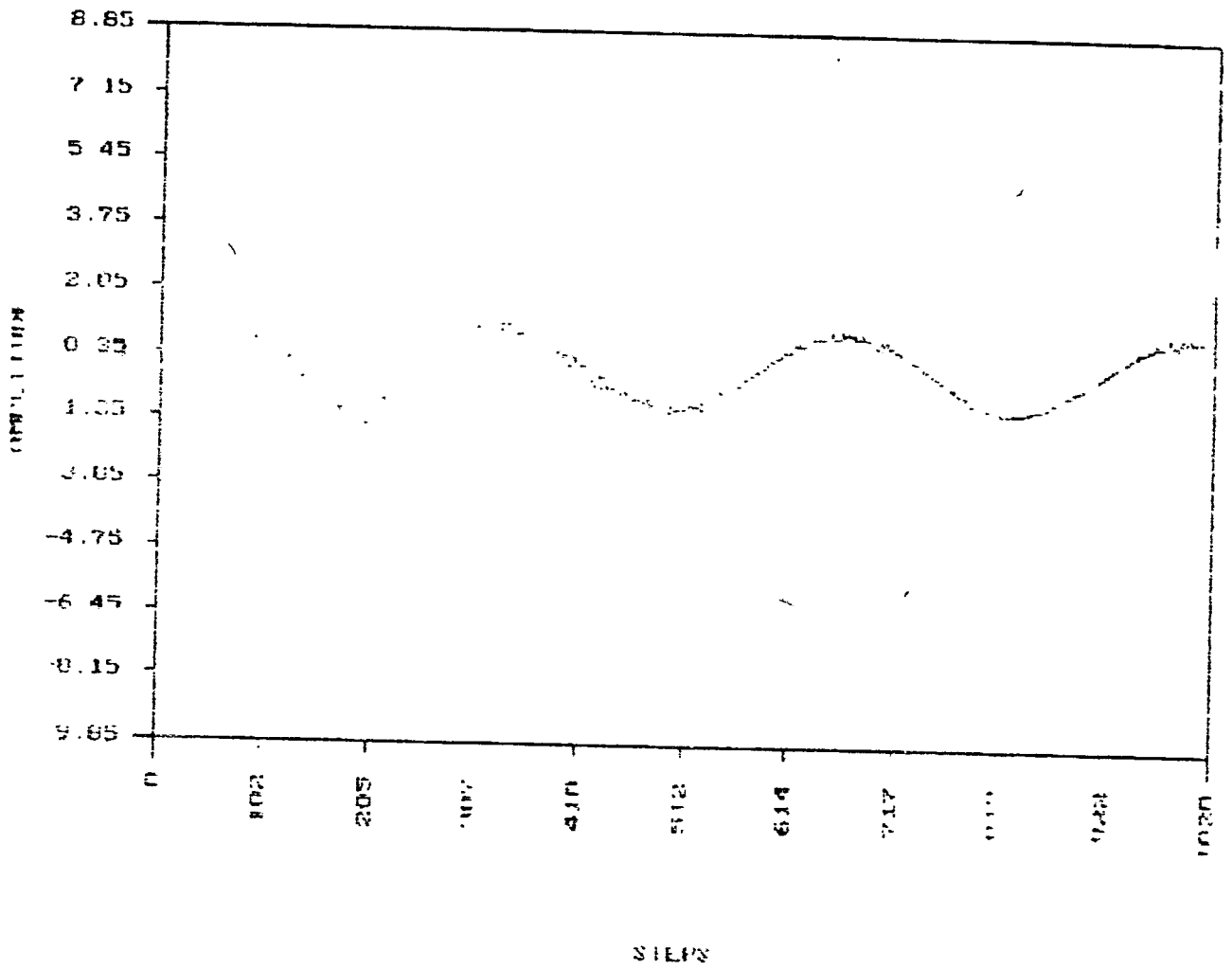
รูปที่ 5.9 สัญญาณเอาต์พุตจากวงจรถ่ายสัญญาณรบกวน  $n_1/n_0 = 1$

MEAN SQUARE ERROR (N1=1)



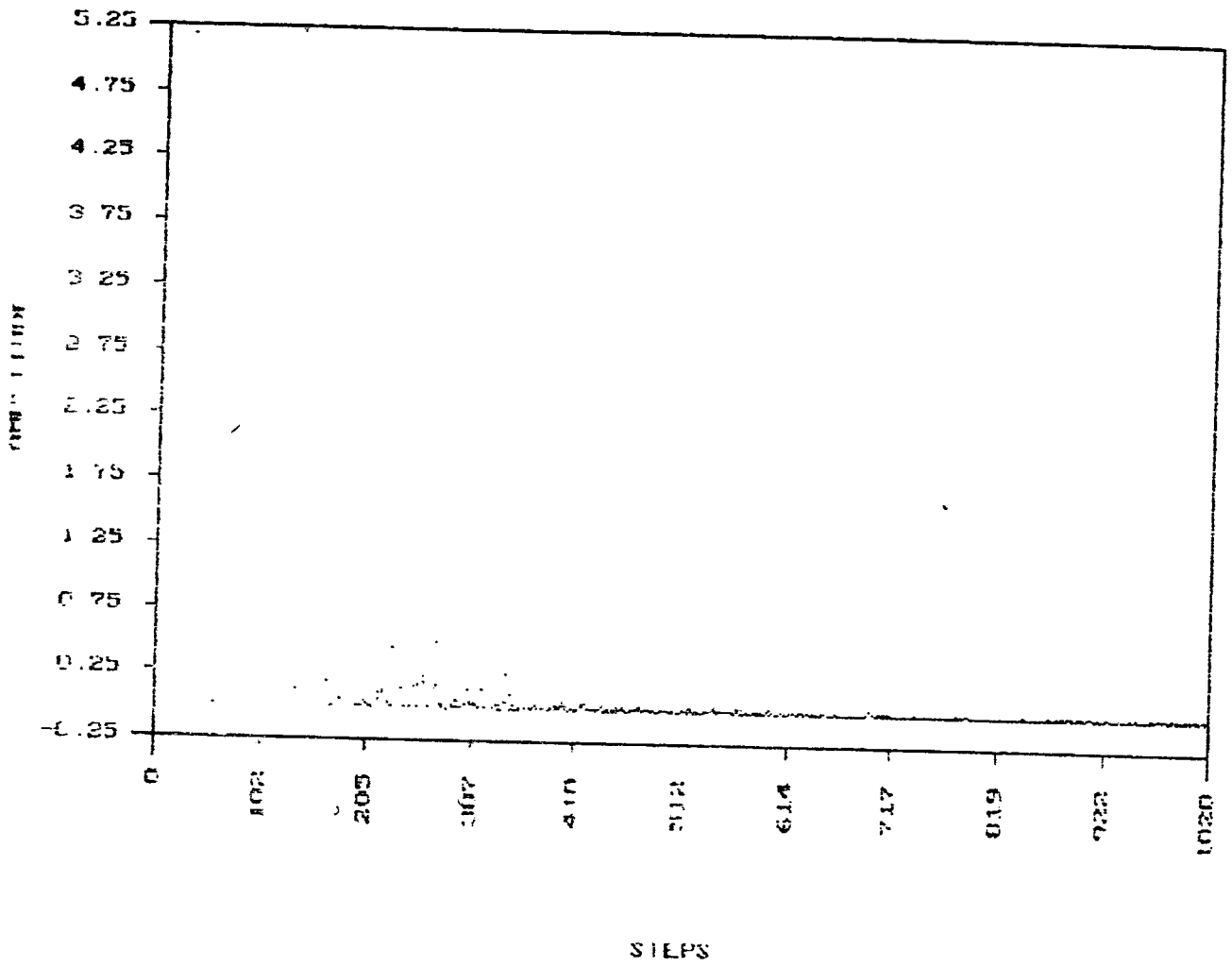
รูปที่ 5.10 กำลังงานของสัญญาณรบกวน  $n_1/n_0 = 1$





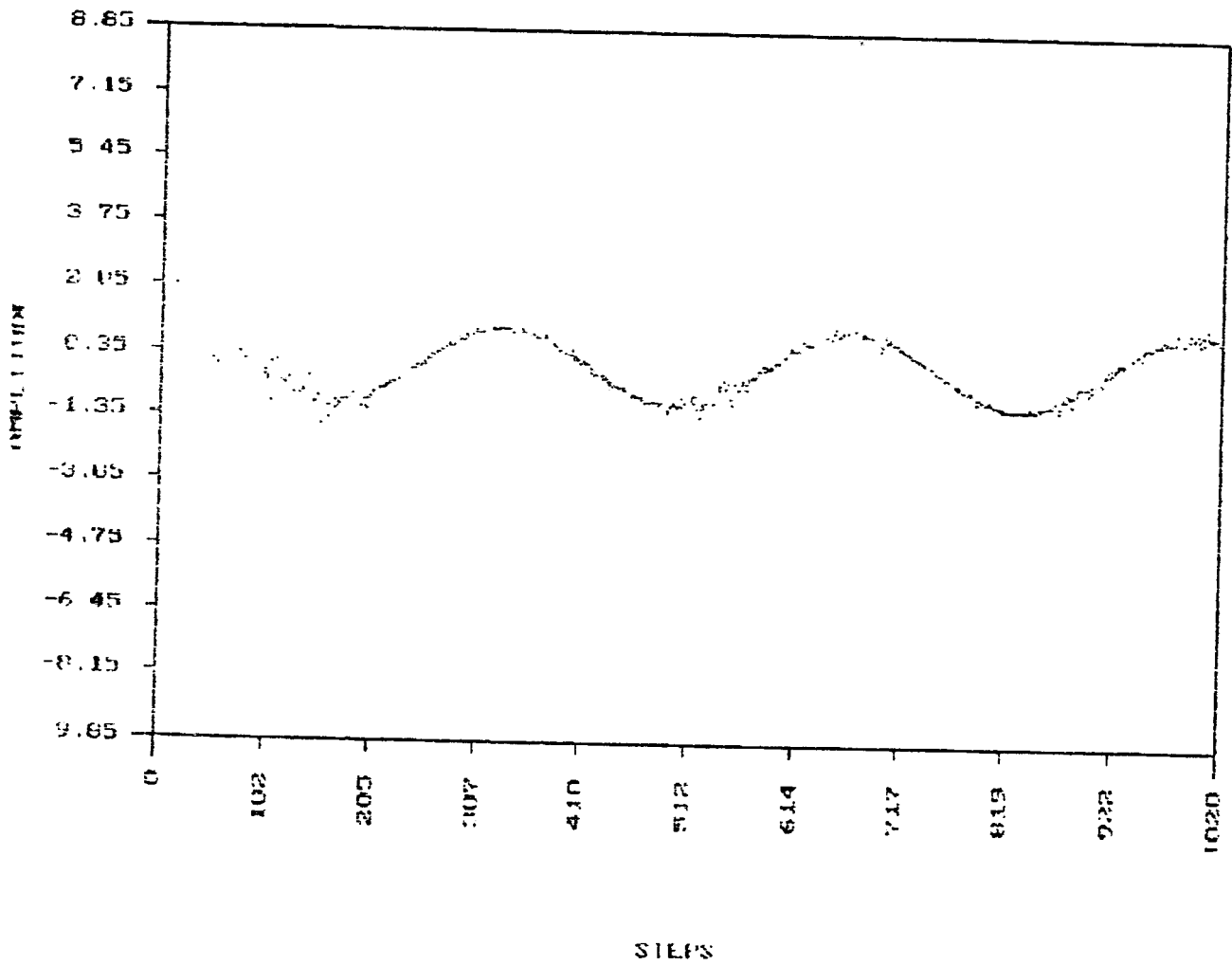
รูปที่ 5.11 สัญญาณเอาต์พุตจากวงจรถักสัญญาณรบกวน  $n_1/n_0 = 2$

## MEAN SQUARE ERROR (N1=2)



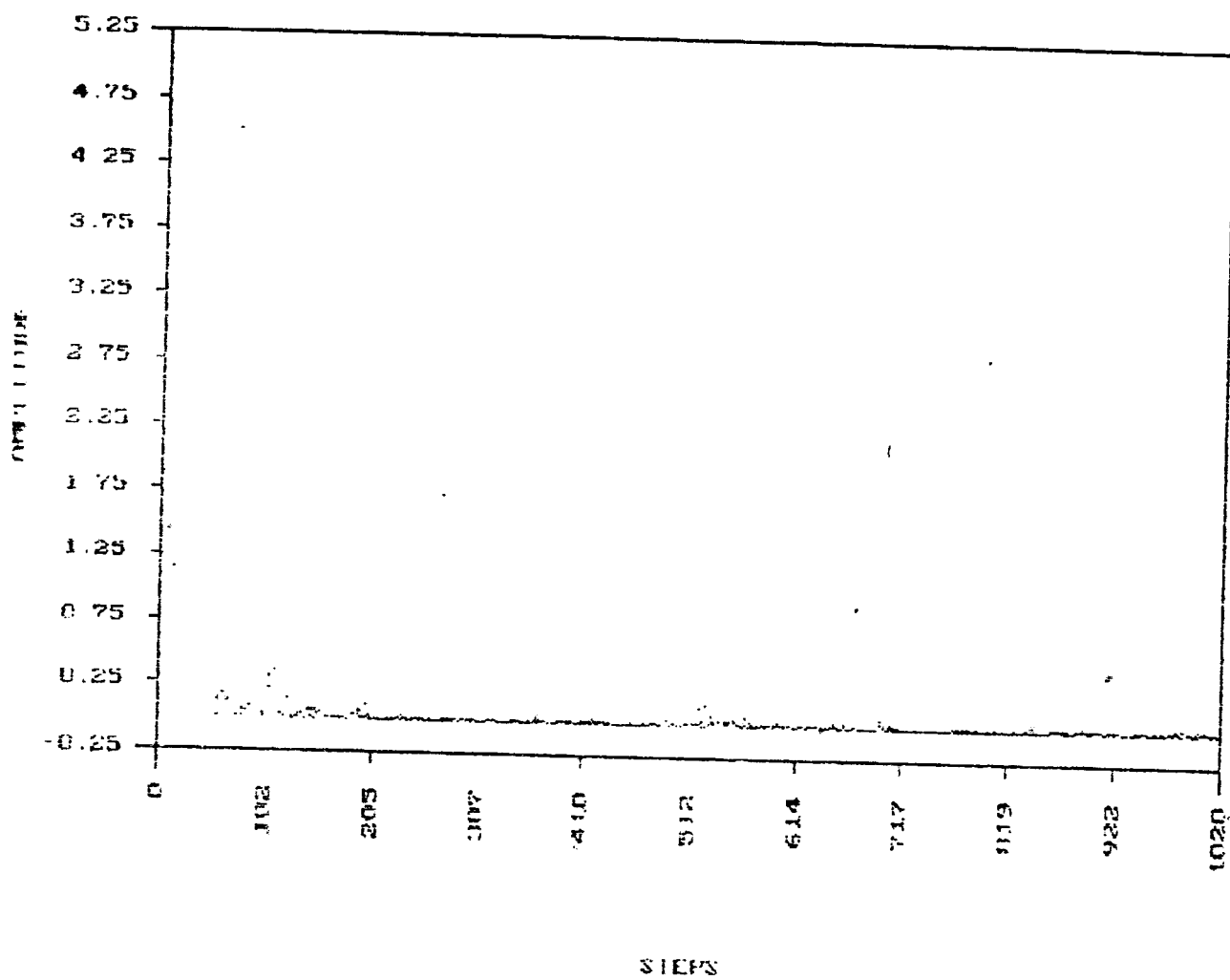
รูปที่ 5.12 กำลังงานของสัญญาณรบกวน  $n_1/n_0 = 2$

REFERENCE INPUT (N1)=3



รูปที่ 5.13 สัญญาณเอาต์พุตจากวงจรถัก้าจัดสัญญาณรบกวน  $n_1/n_0 = 3$

## MEAN SQUARE ERROR (N1=3)



รูปที่ 5.14 กำลังงานของสัญญาณรบกวน  $n_1/n_0 = 3$

### 5.1.3 การทดลองที่ 3 : คุณผลจากการแปรค่า สัญญาณรบกวน ( $n_o$ )

เงื่อนไขคงที่ทุกกรณี :

ขนาดสัญญาณอินพุตอ้างอิง  $n_1 = 1$

พารามิเตอร์ของการลู่เข้า  $\mu = 0.0005$

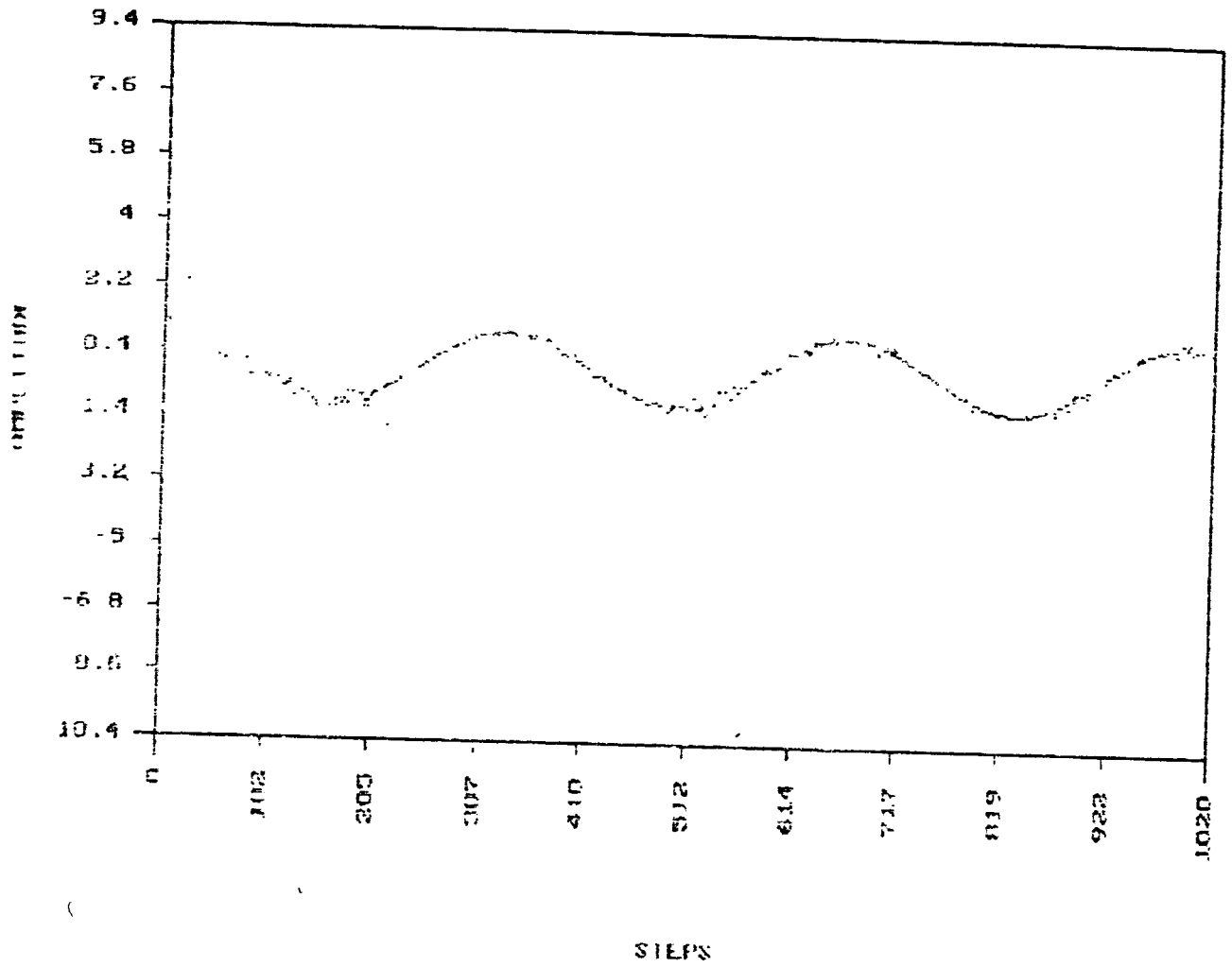
ตัวแปร : ขนาดของสัญญาณรบกวน  $n_o$  ในอินพุตหลัก

ได้ผลลัพธ์แสดงในรูปที่ 5.15 ถึงรูปที่ 5.20 และสรุปผลแสดงในตารางที่ 5.3

$n_o/s$	steps	S/N (dB)
1	220	10.385
3	200	2.106
6	150	-3.686

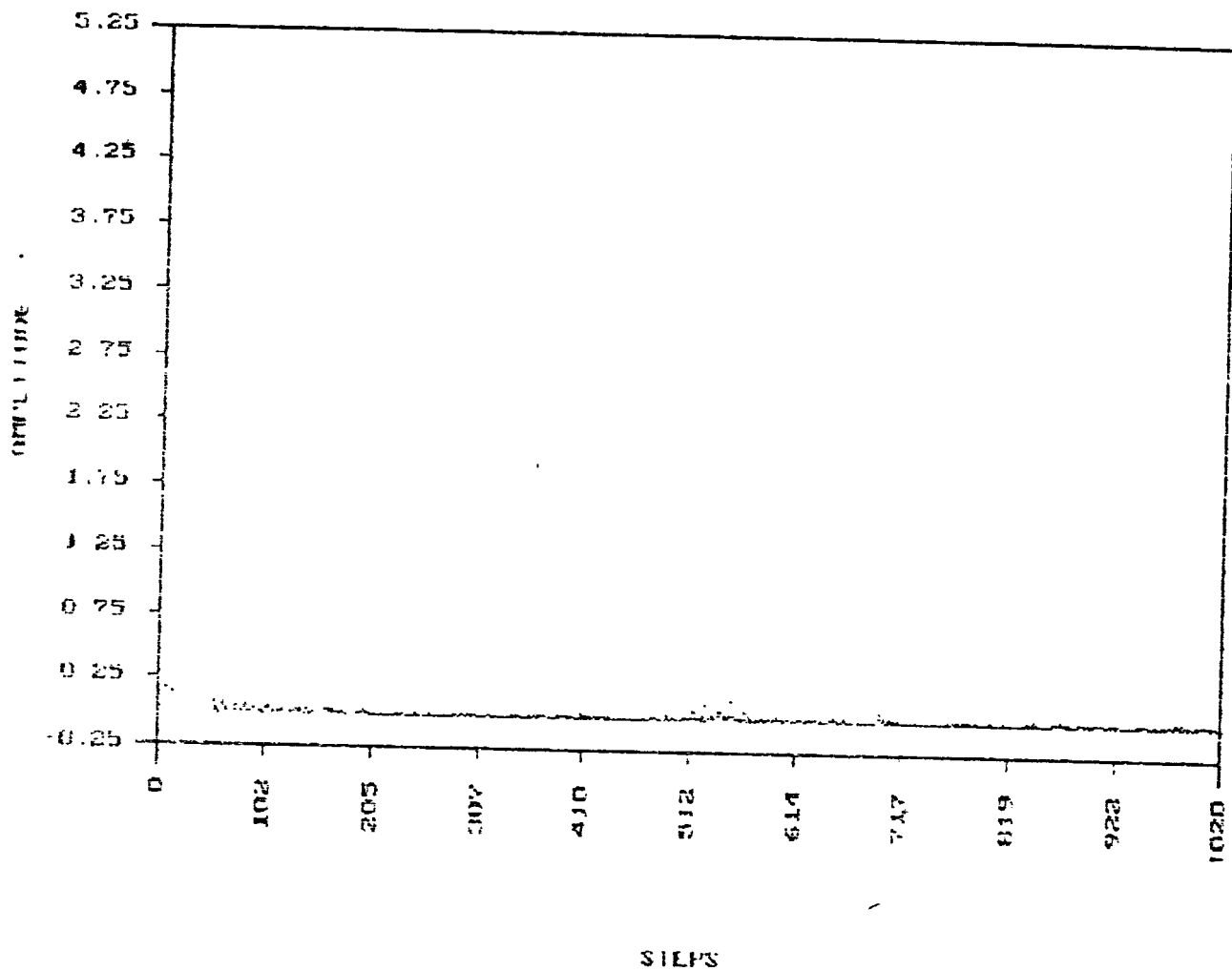
ตารางที่ 5.3

NOISE/SIGNAL = 1



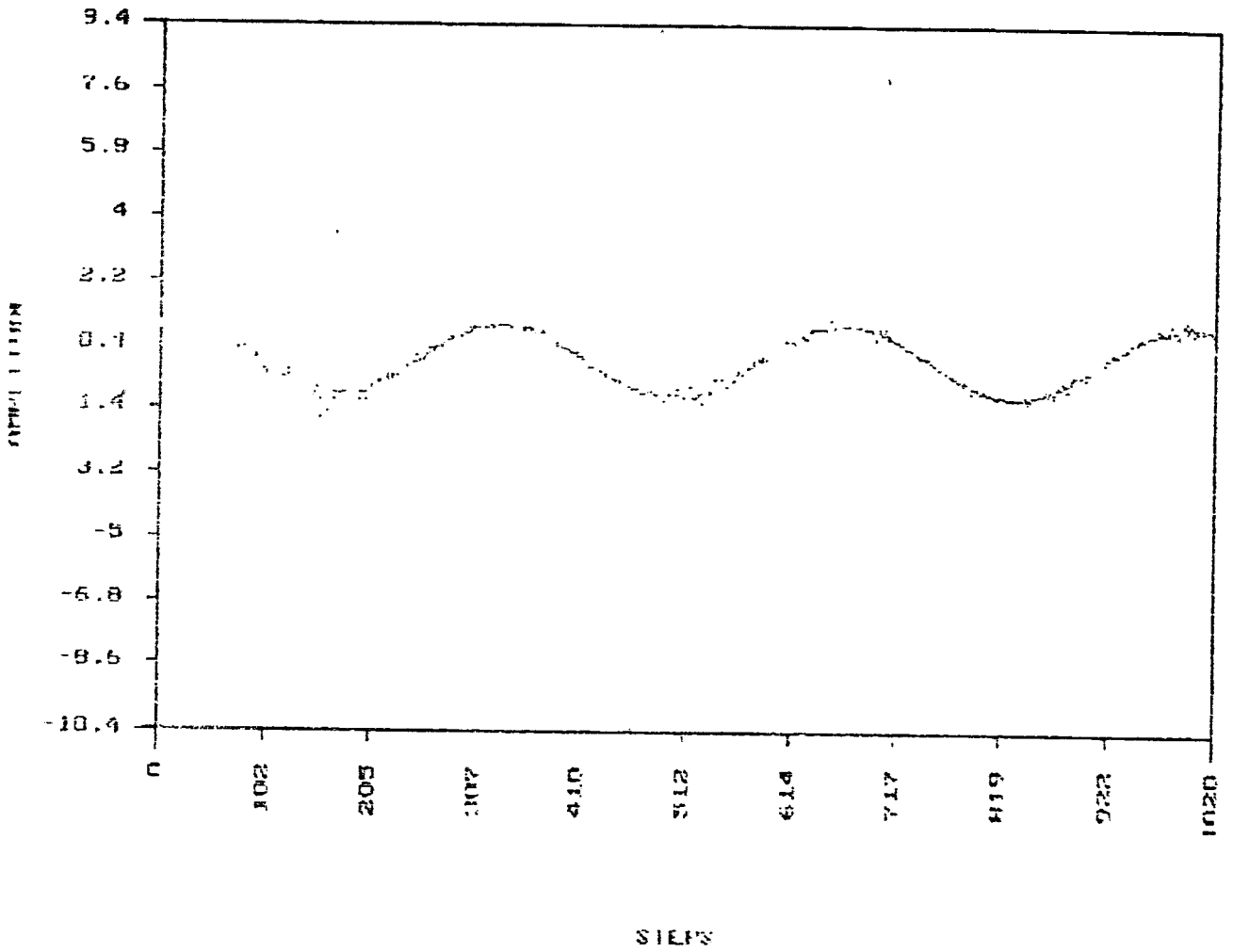
รูปที่ 5.15 สัญญาณเอากัฟจากวงจรถักำจัดสัญญาณรบกวน  $n_0/s = 1$

MEAN SQUARE ERROR (NO/S=1)



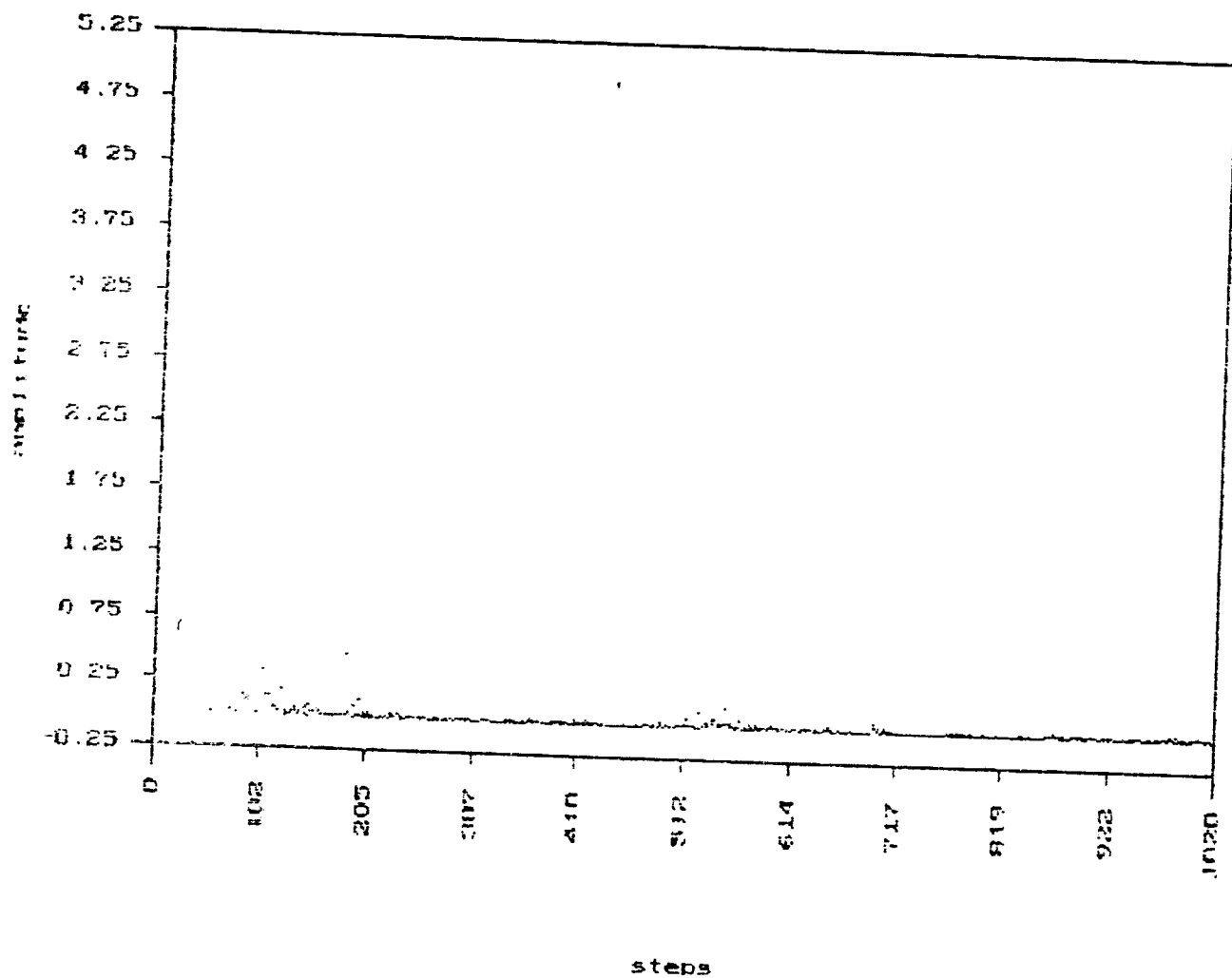
รูปที่ 5.16 กำลังงานของสัญญาณรบกวน  $n_0/s = .1$

NOISE/SIGNAL =3



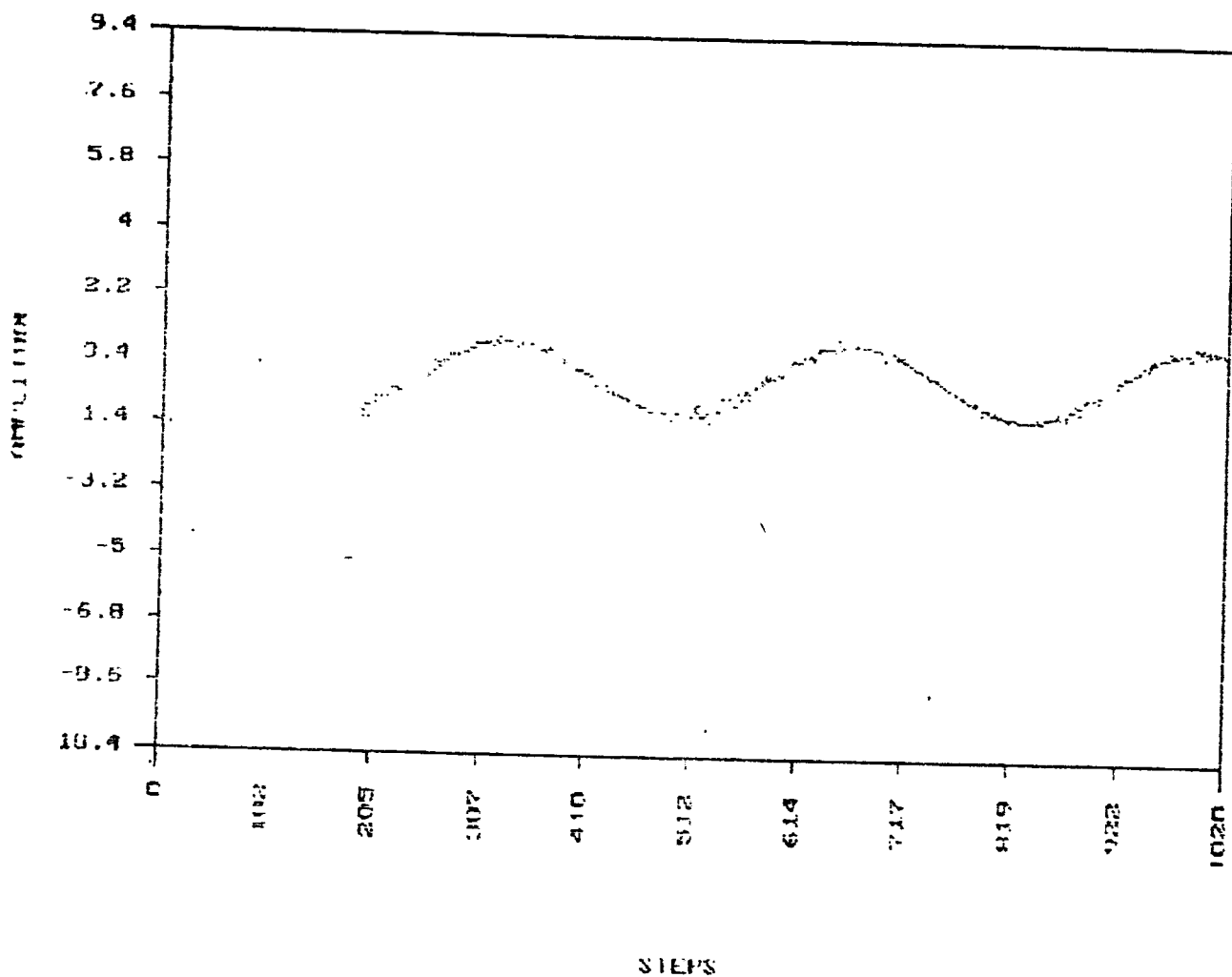
รูปที่ 5.17 สัญญาณเอาร์ทพุทจากวงจรถักัดสัญญาณรบกวน  $n_0/s = 3$

## MEAN SQUARE ERROR (NO./S=3)



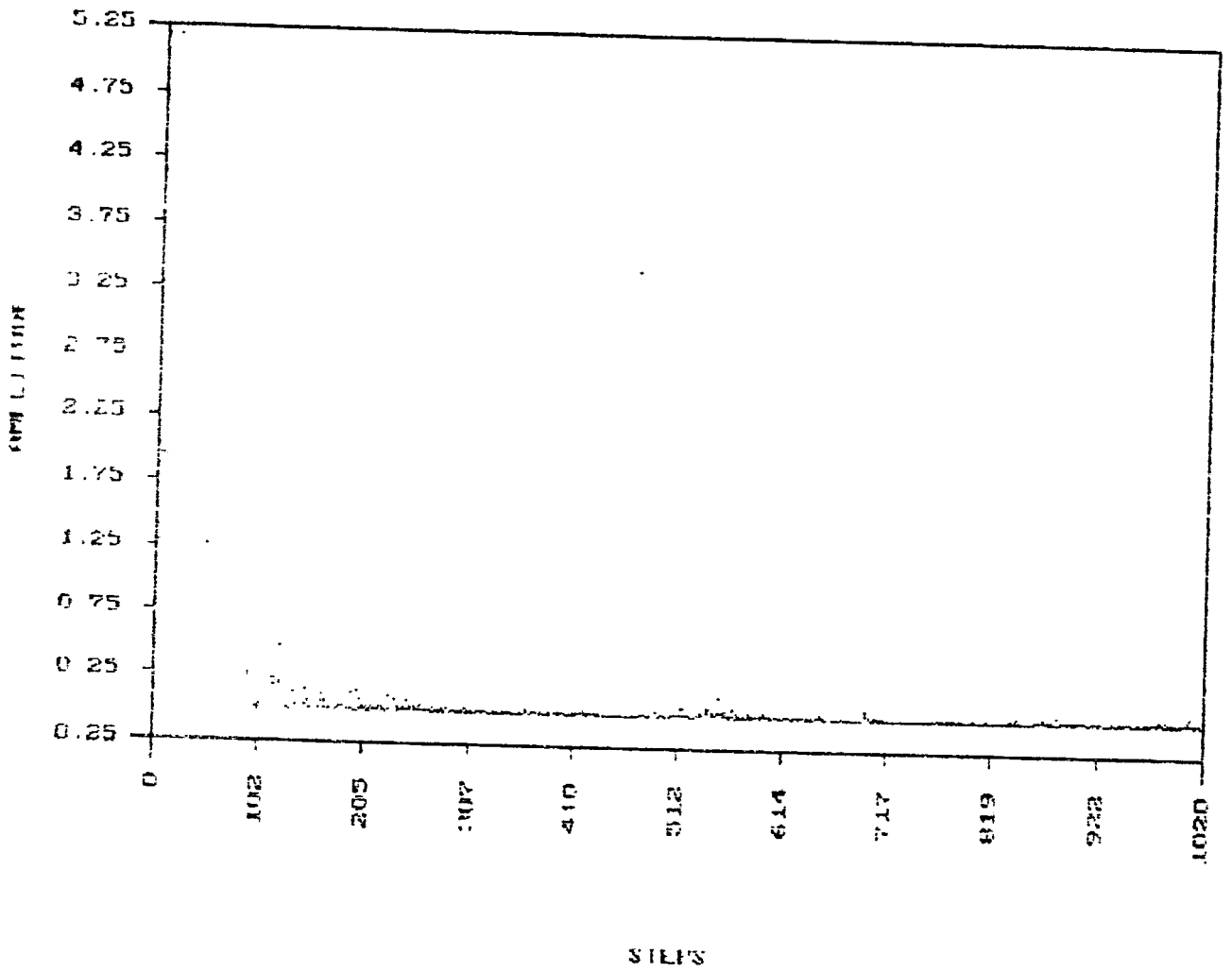
รูปที่ 5.18 กำลังงานของสัญญาณรบกวน  $n_0/s = 3$

NOISE/SIGNAL = 6



รูปที่ 5.19 สัญญาณเอวท์พุกจากวงจรกำจัดสัญญาณรบกวน  $n_0/s = 6$

MEAN SQUARE ERROR



รูปที่ 5.20 กำลังงานของสัญญาณรบกวน  $n_0/s = 6$

## 5.2 สัญญาณที่ต้องการ (s) มีองค์ประกอบของความถี่หลายความถี่

ลักษณะของสัญญาณที่ต้องการ มีความถี่ผสมกัน 3 ความถี่ โดยให้ ค่าอัตราส่วนความถี่ของสัญญาณ ต่อความถี่สัญญาณสุ่มมี 3 ค่าคือ 0.003, 0.005 และ 0.007 แสดงในรูปที่ 5.21 และเมื่อมีสัญญาณรบกวนมาผสมกับสัญญาณนี้ โดยมี S/N เท่ากับ -7.996 dB ได้ลักษณะรูปสัญญาณ ดังรูปที่ 5.22

### 5.2.1 การทดลองที่ 4 : ผลจากการแปรค่าพารามิเตอร์ของการลู่เข้า ( $\mu$ )

เงื่อนไขคงที่ทุกกรณี :

$$\text{สัญญาณอินพุตหลัก} = s+n_0 \quad \text{โดยที่ } s/n_0 = 1$$

$$\text{สัญญาณอินพุตอ้างอิง} = n_1 \quad \text{โดยที่ } n_1 = n_0$$

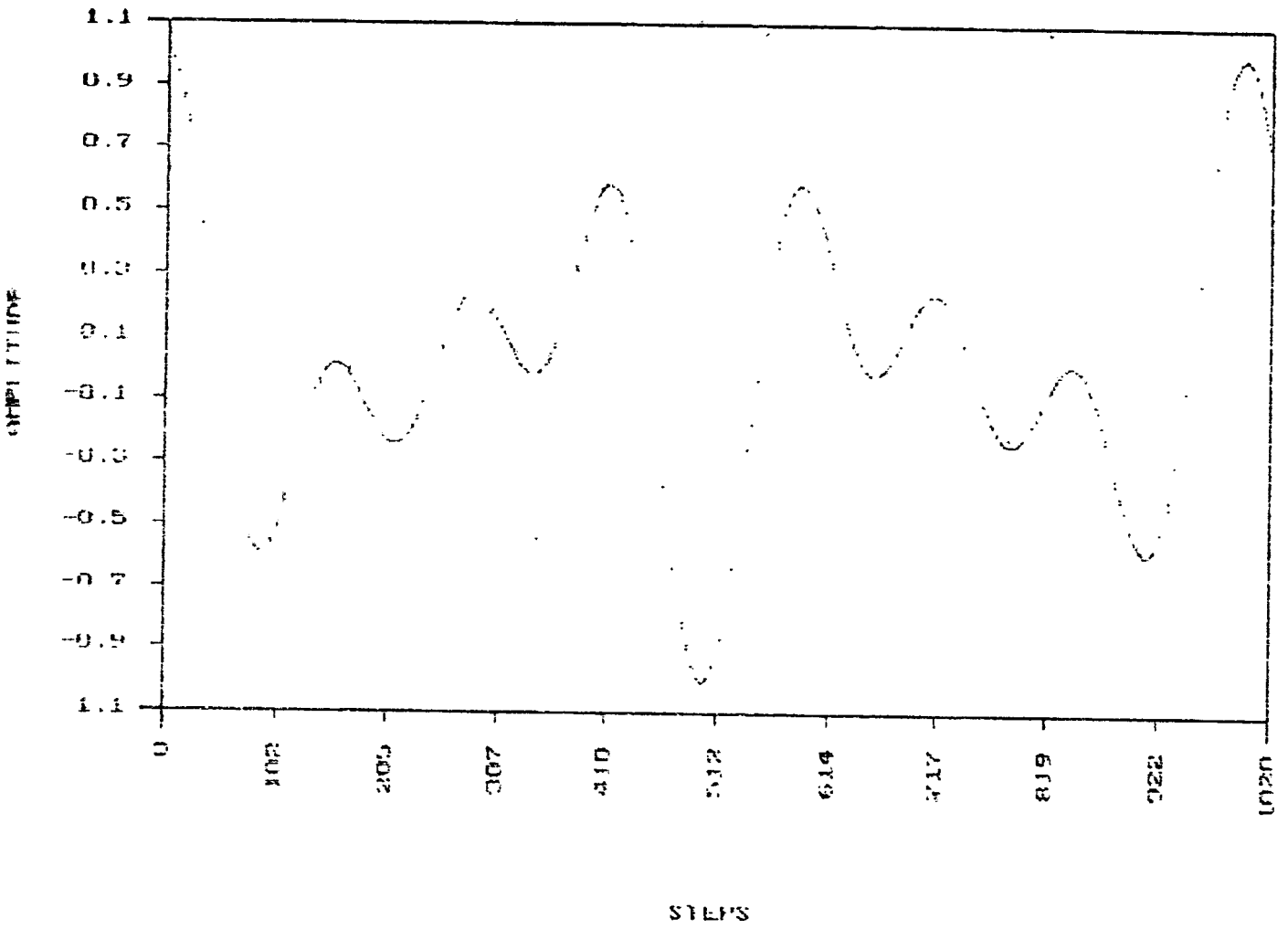
ตัวแปร : ค่าพารามิเตอร์ของการลู่เข้า  $\mu$

ได้ผลลัพธ์แสดงใน รูปที่ 5.23 ถึงรูปที่ 5.30 นำข้อมูลมาวิเคราะห์ แสดงในตารางที่ 5.4

$\mu$	steps	S/N (dB)
.003	400	3.153
.005	250	5.098
.008	150	6.704
.010	125	7.353

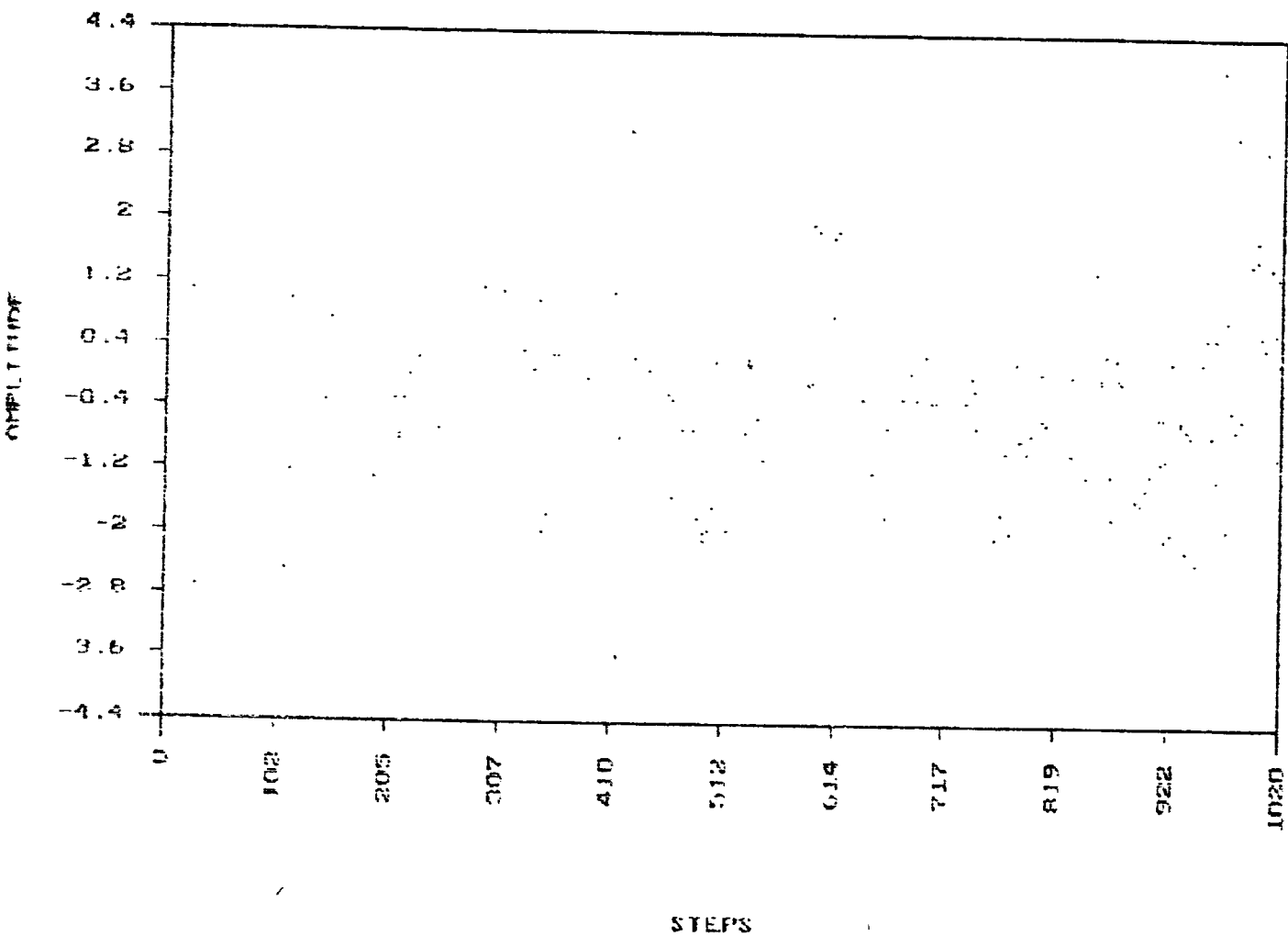
ตารางที่ 5.4

SIGNAL

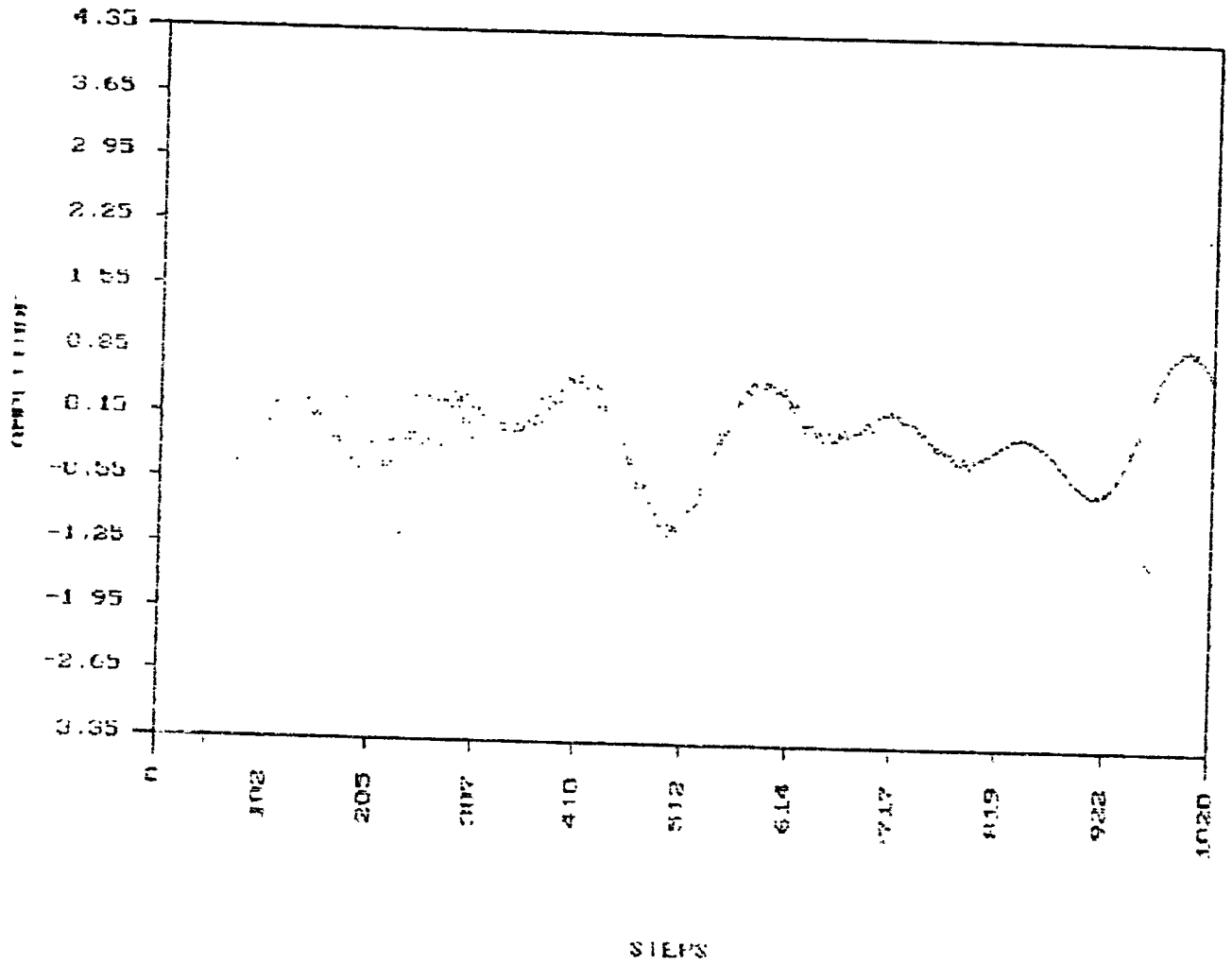


รูปที่ 5.21 สัญญาณที่ต้องการ (s)

## SIGNAL + NOISE

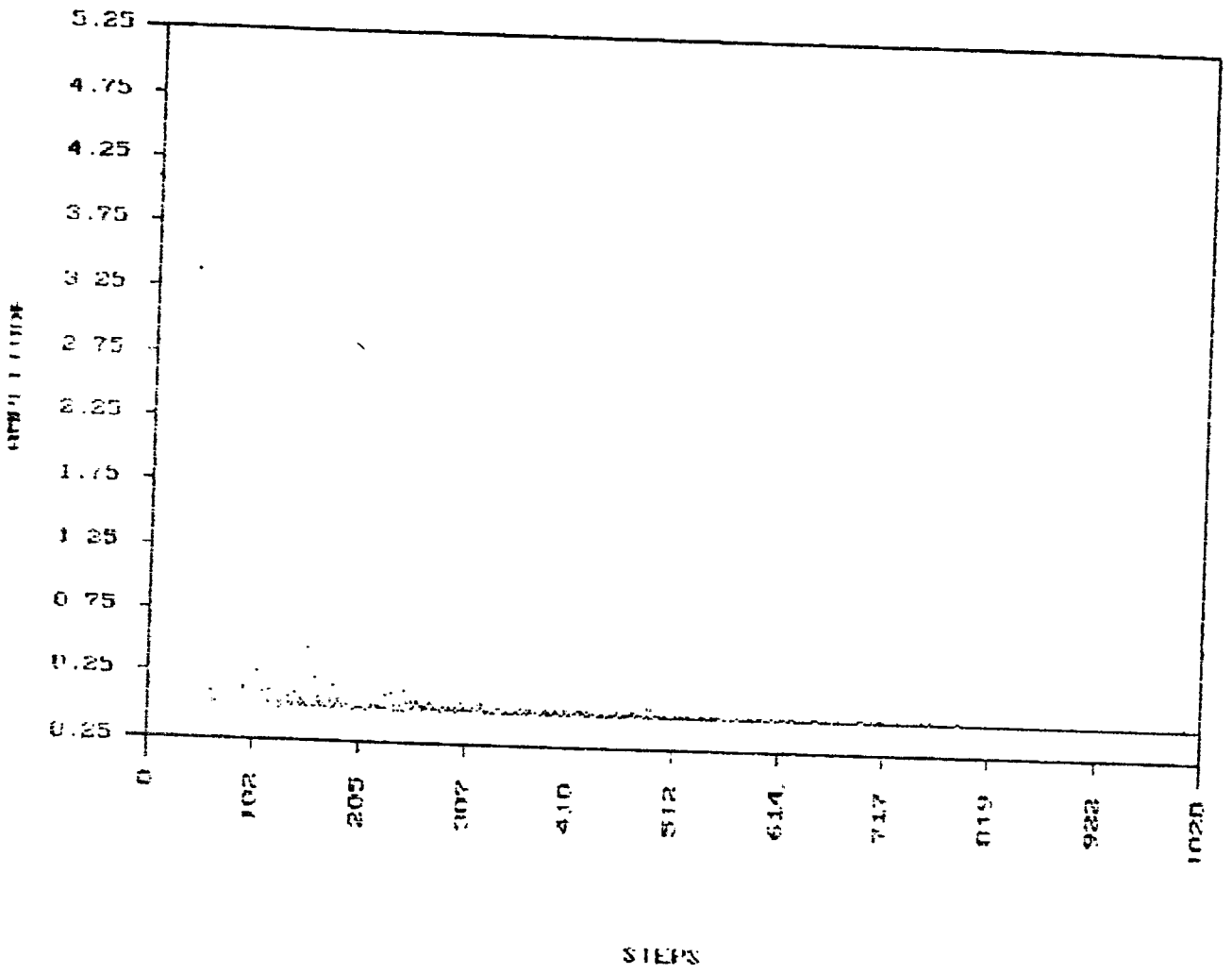


รูปที่ 5.22 สัญญาณอินพุทหลัก

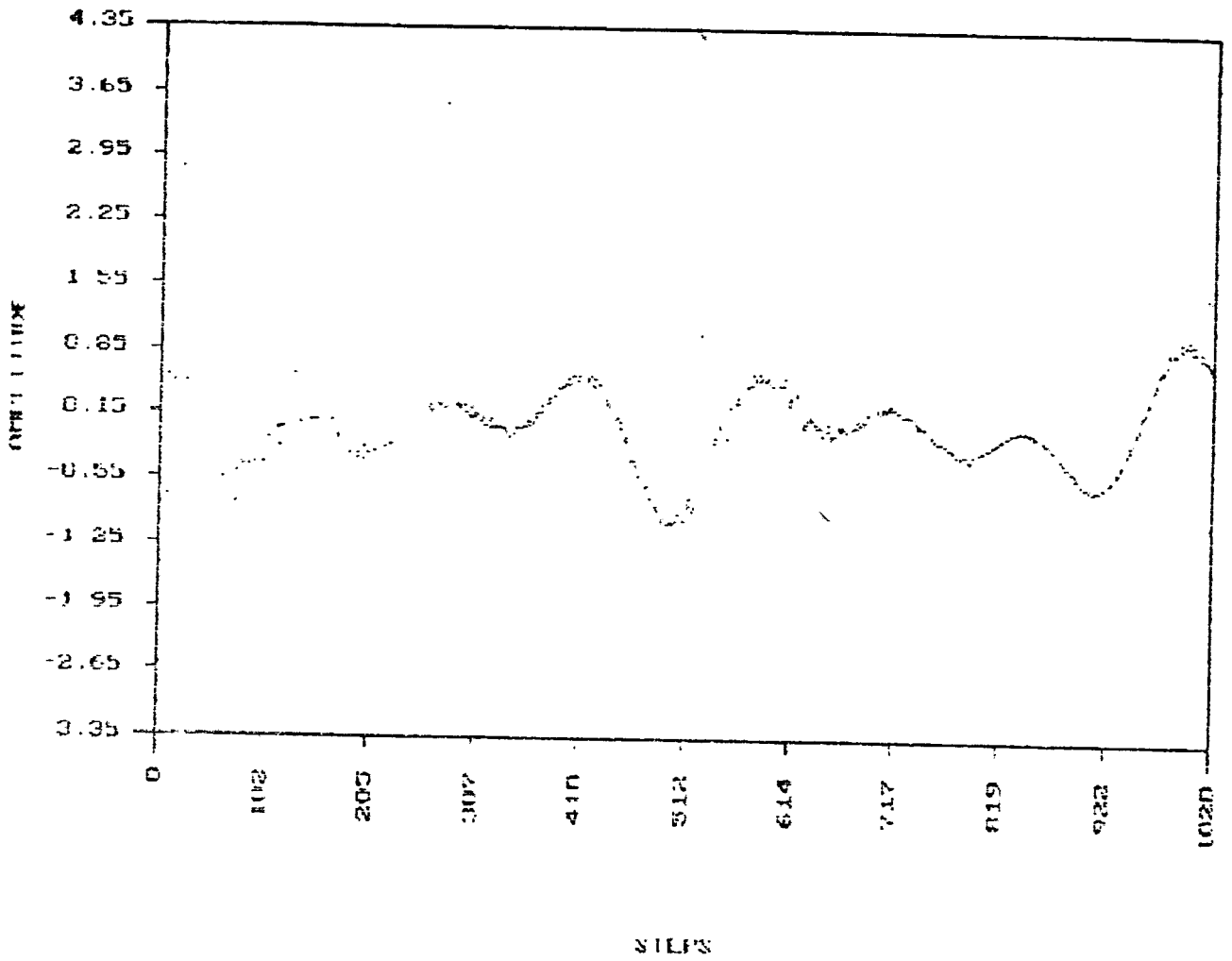
$\mu = 0.003$ 


รูปที่ 5.23 สัญญาณเอาต์พุตจากวงจรถ่ายสัญญาณรบกวน  $\mu = 0.003$

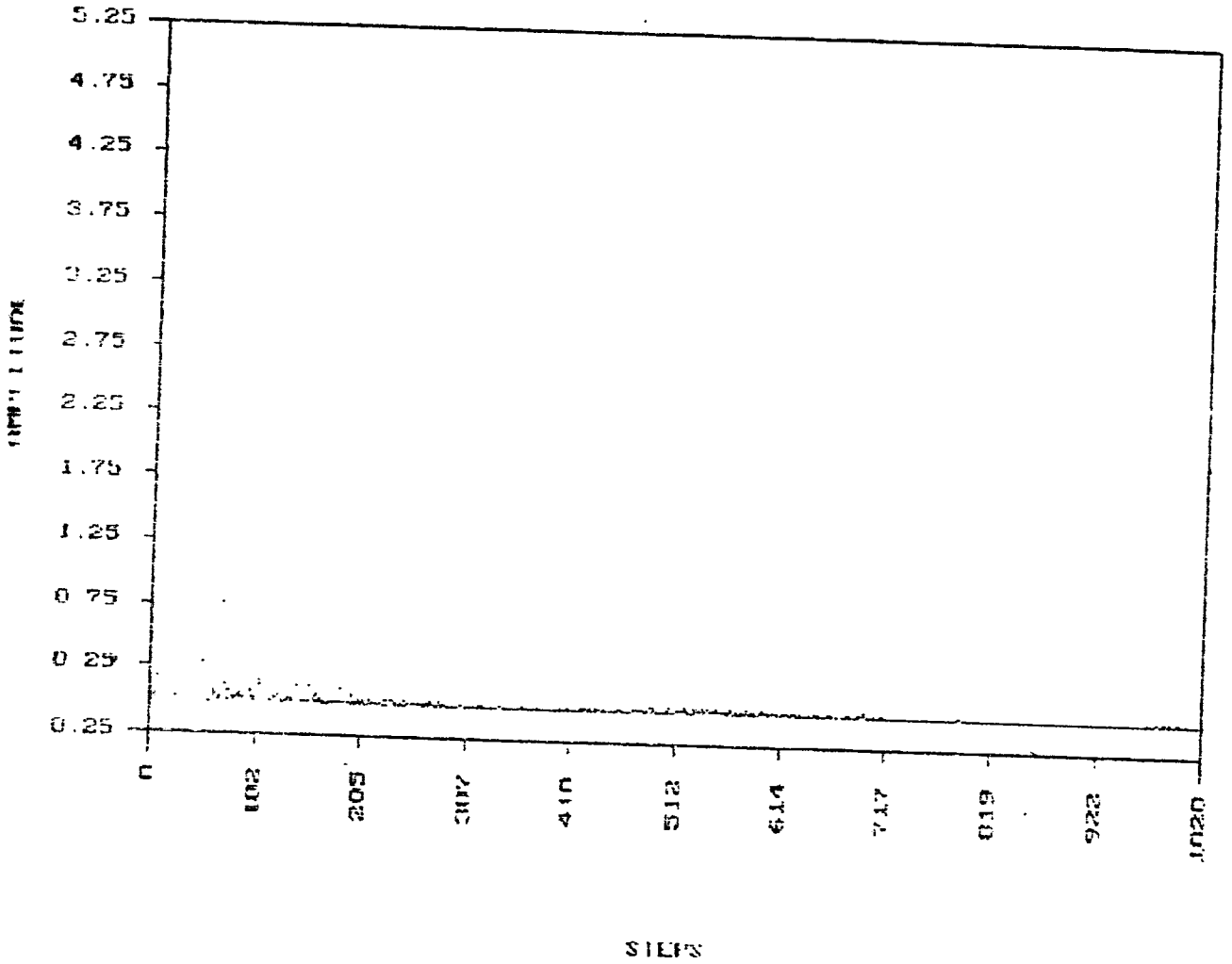
MEAN SQUARE ERROR ( $\mu=0.003$ )



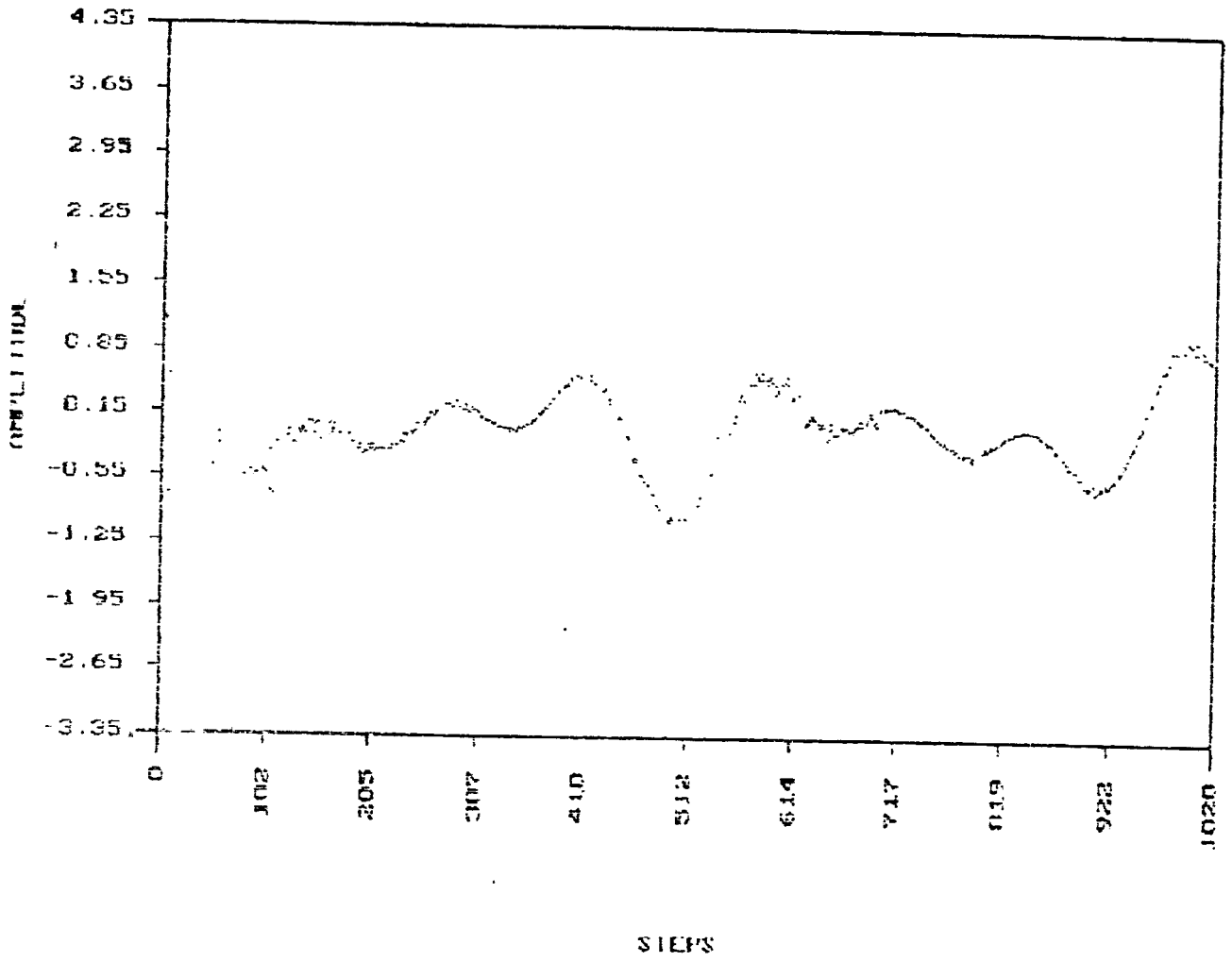
รูปที่ 5.24 กำลังงานของสัญญาณรบกวน  $\mu = 0.003$



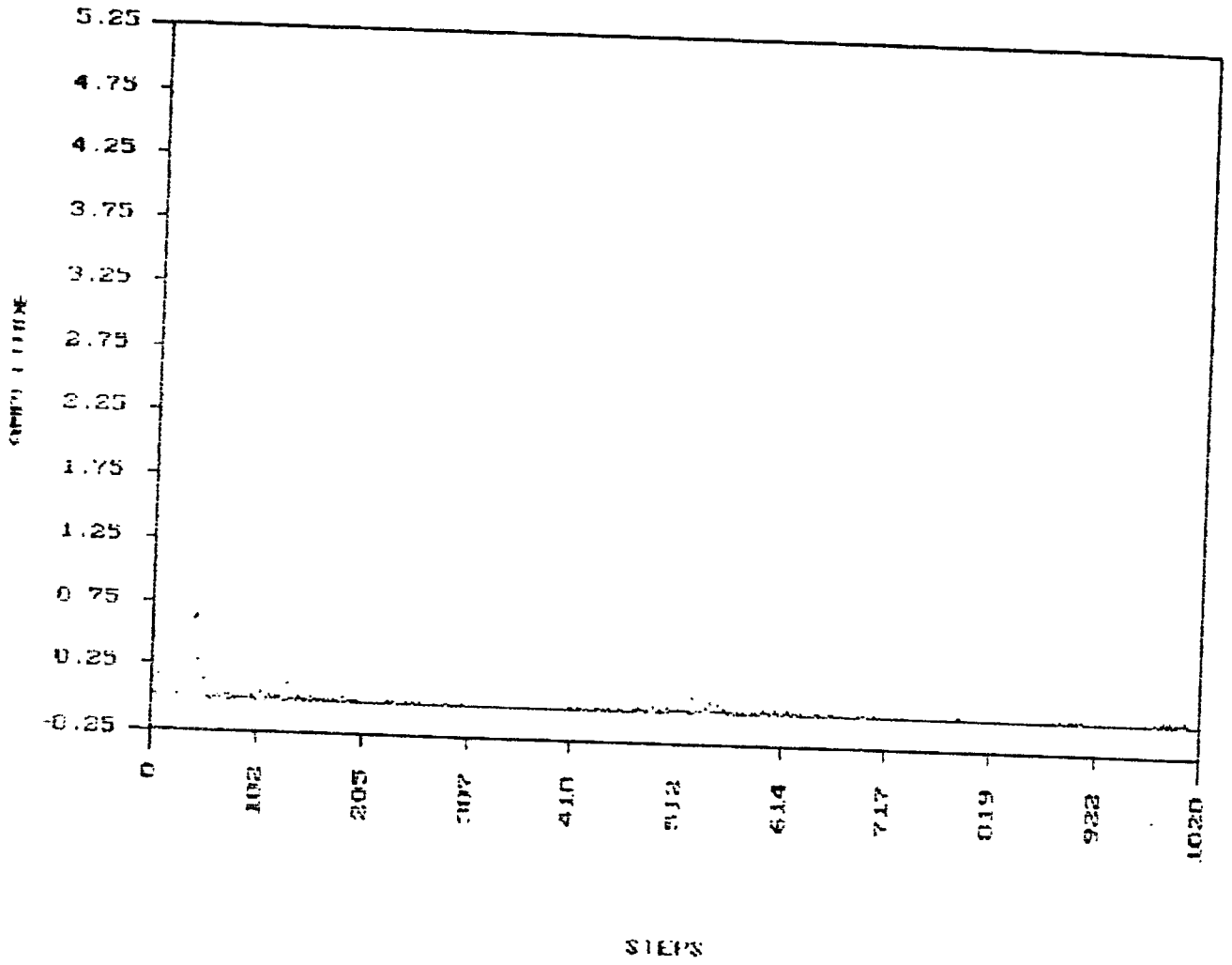
รูปที่ 5.25 สัญญาณเออร์เกนที่พ่วงจากวงจรกำจัดสัญญาณรบกวน  $\mu = 0.005$



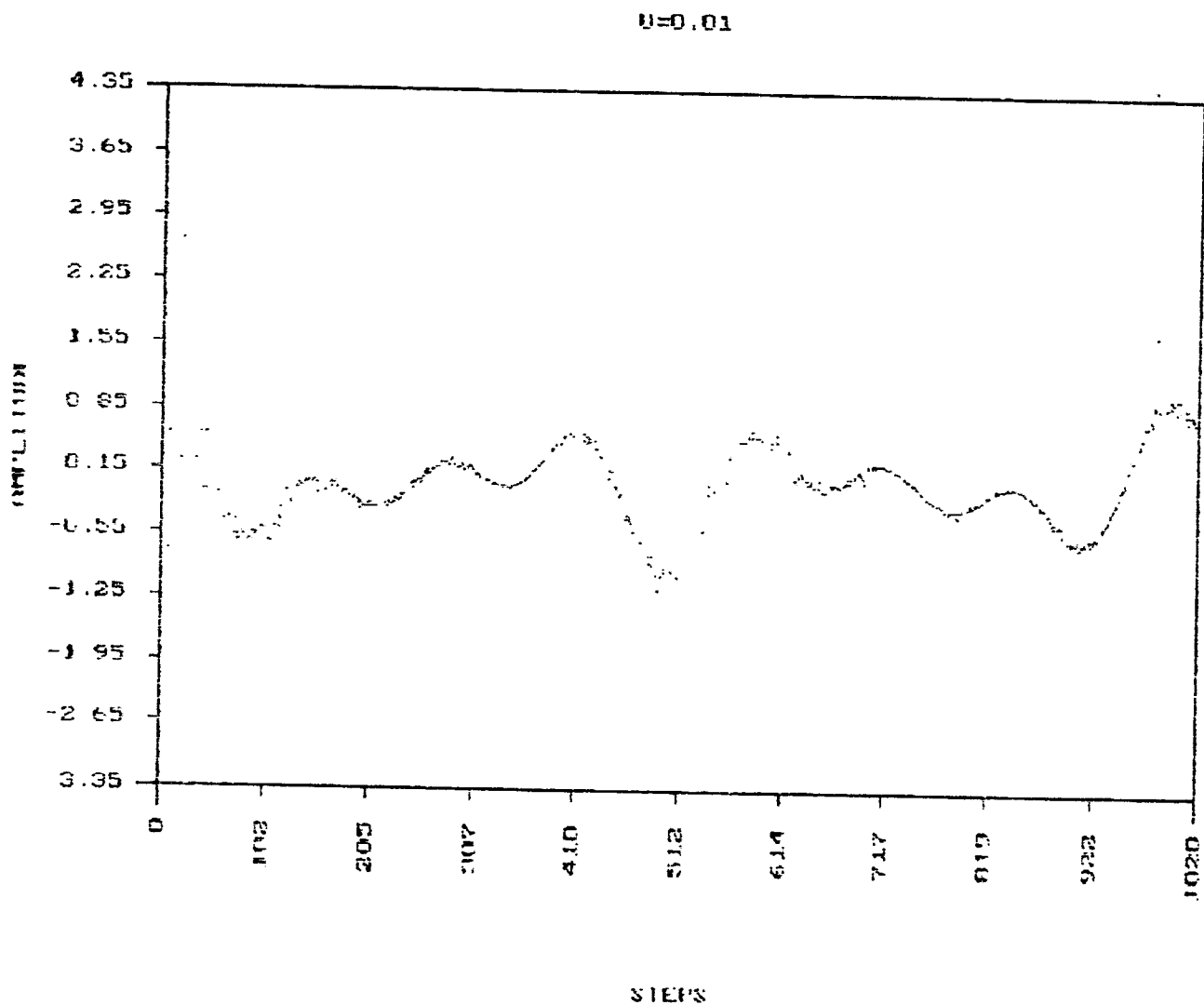
รูปที่ 5.26 กำลังงานของสัญญาณรบกวน  $\mu = 0.005$



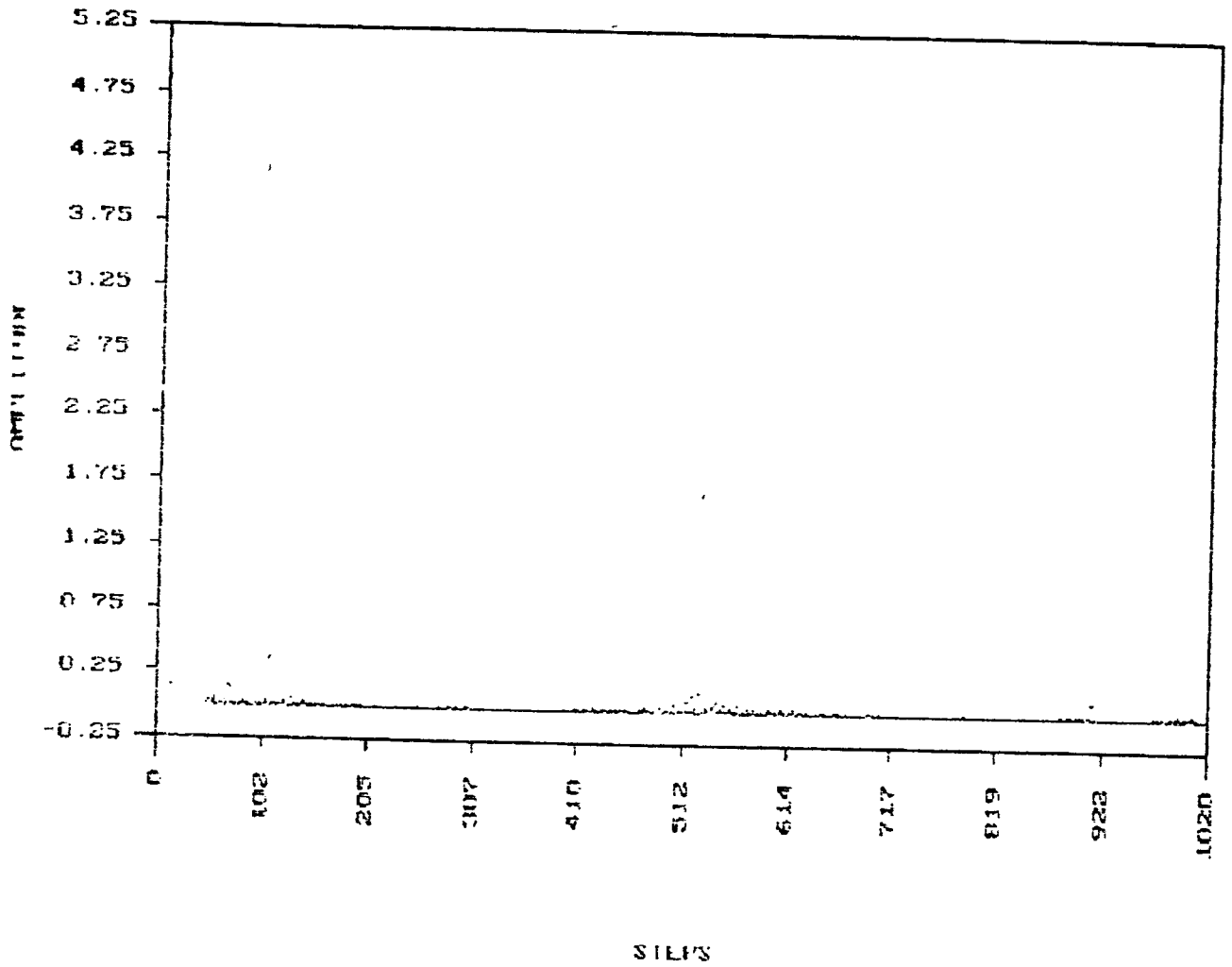
รูปที่ 5.27 สัญญาณเอากัฟจากวงจรกำจัดสัญญาณรบกวน  $\mu = 0.008$



รูปที่ 5.28 กำลังงานของสัญญาณรบกวน  $\mu = 0.008$



รูปที่ 5.29 สัญญาณเออร์เกนที่พุดจากวงจรถัก้าจัดสัญญาณรบกวน  $\mu = 0.010$

MEAN SQUARE ERROR ( $U=0.01$ )

รูปที่ 5.30 กำลังงานของสัญญาณรบกวน  $\mu = 0.010$

### 5.2.2 การทดลองที่ 5 : ผลจากการแปรค่า ขนาดของอินพุตอ้างอิง

เงื่อนไขคงที่ทุกกรณี :

$$\text{สัญญาณอินพุตหลัก} = s+n_0 \quad \text{โดยที่ } s/n_0 = 0.3333$$

$$\text{พารามิเตอร์ของการลู่เข้า } \mu = 0.001$$

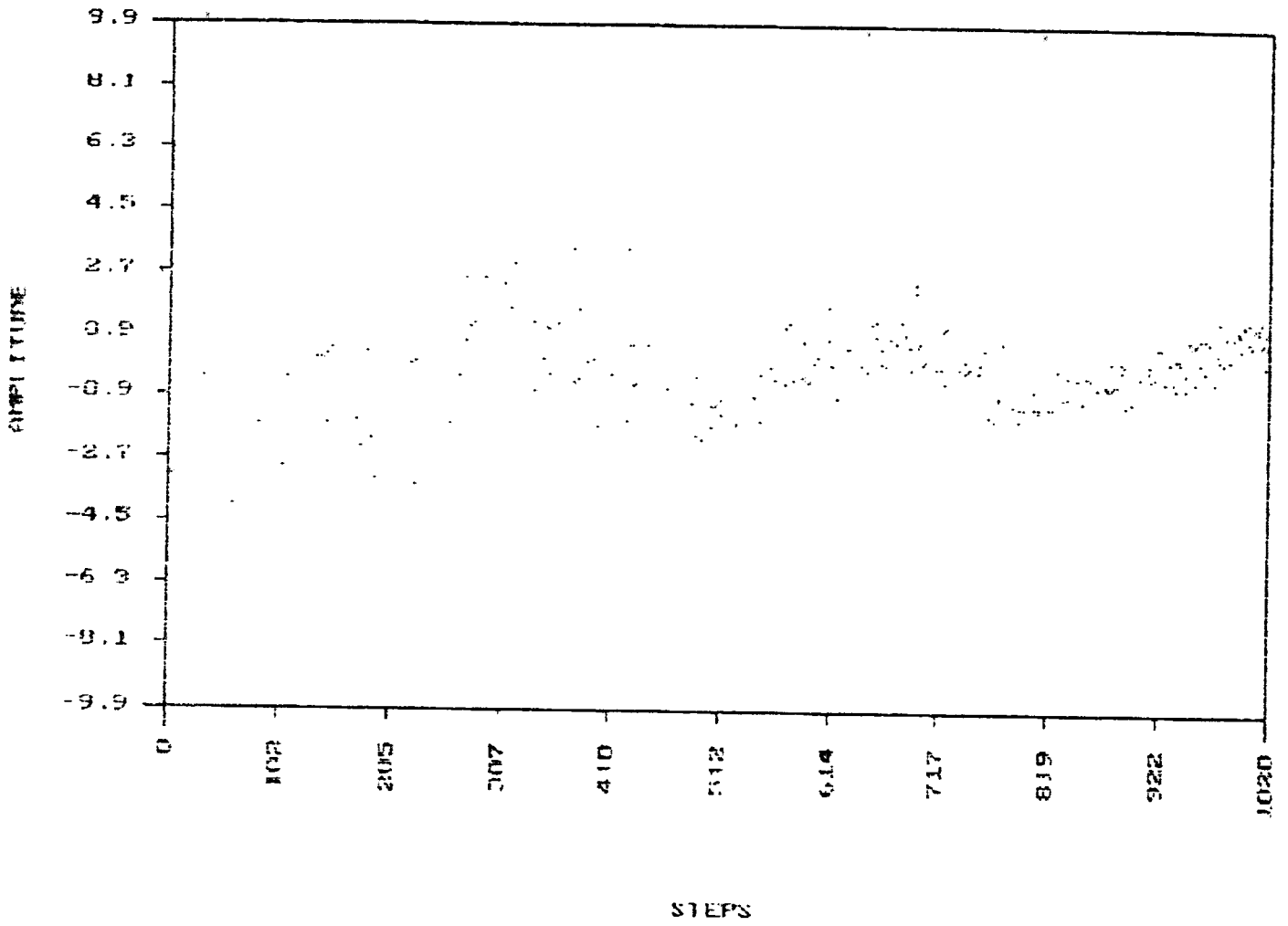
ตัวแปร : ขนาดของอินพุตอ้างอิง  $n_1$

ได้ผลลัพธ์แสดงในรูปที่ 5.31 ถึงรูปที่ 5.38 และสรุปผลแสดงในตารางที่ 5.5

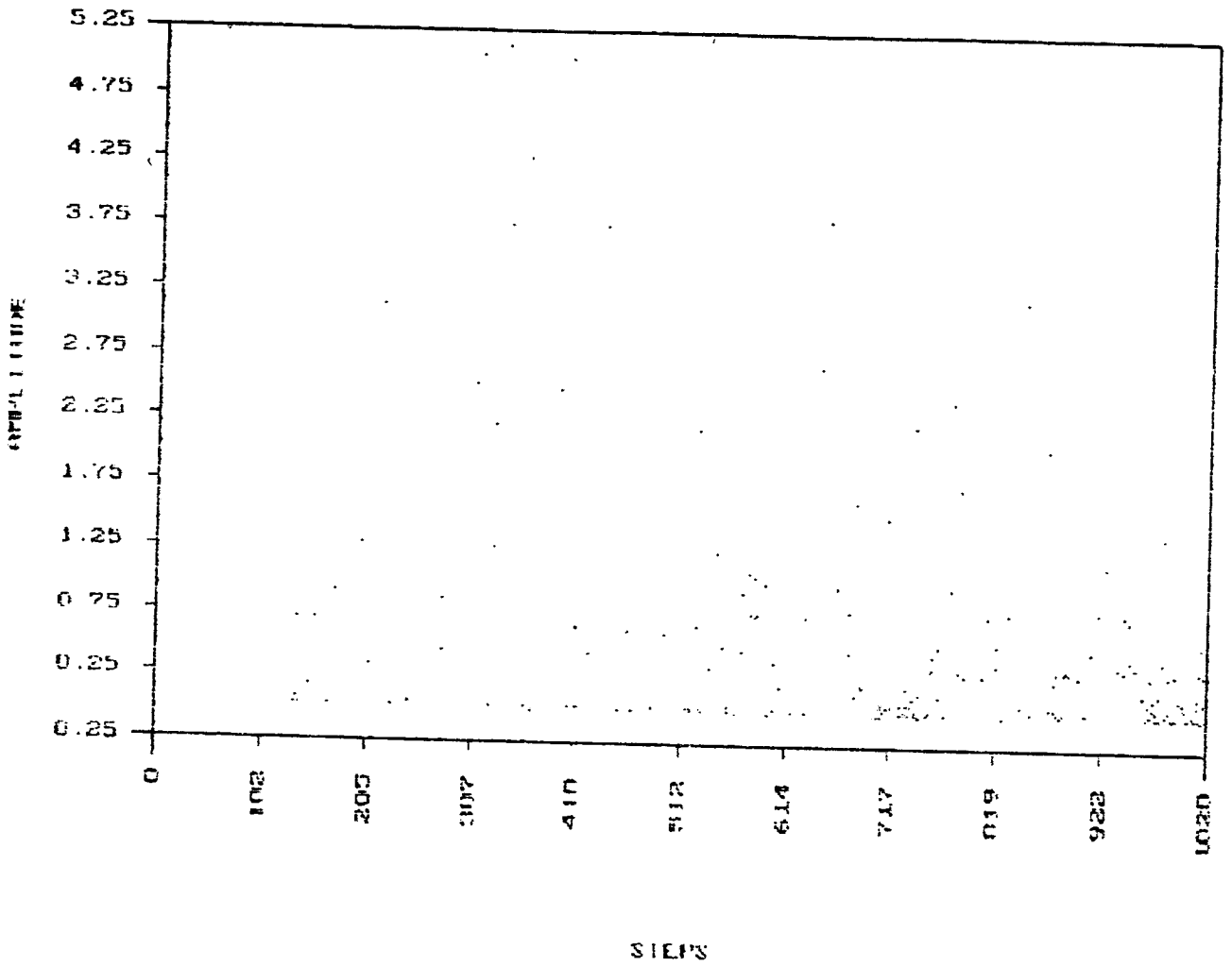
$n_1/n_0$	steps	S/N (db)
1	>1000	-11.30
2	350	-5.0
3	150	-1.72
4	75	0.433

ตารางที่ 5.5

REFERENCE INPUT = 1

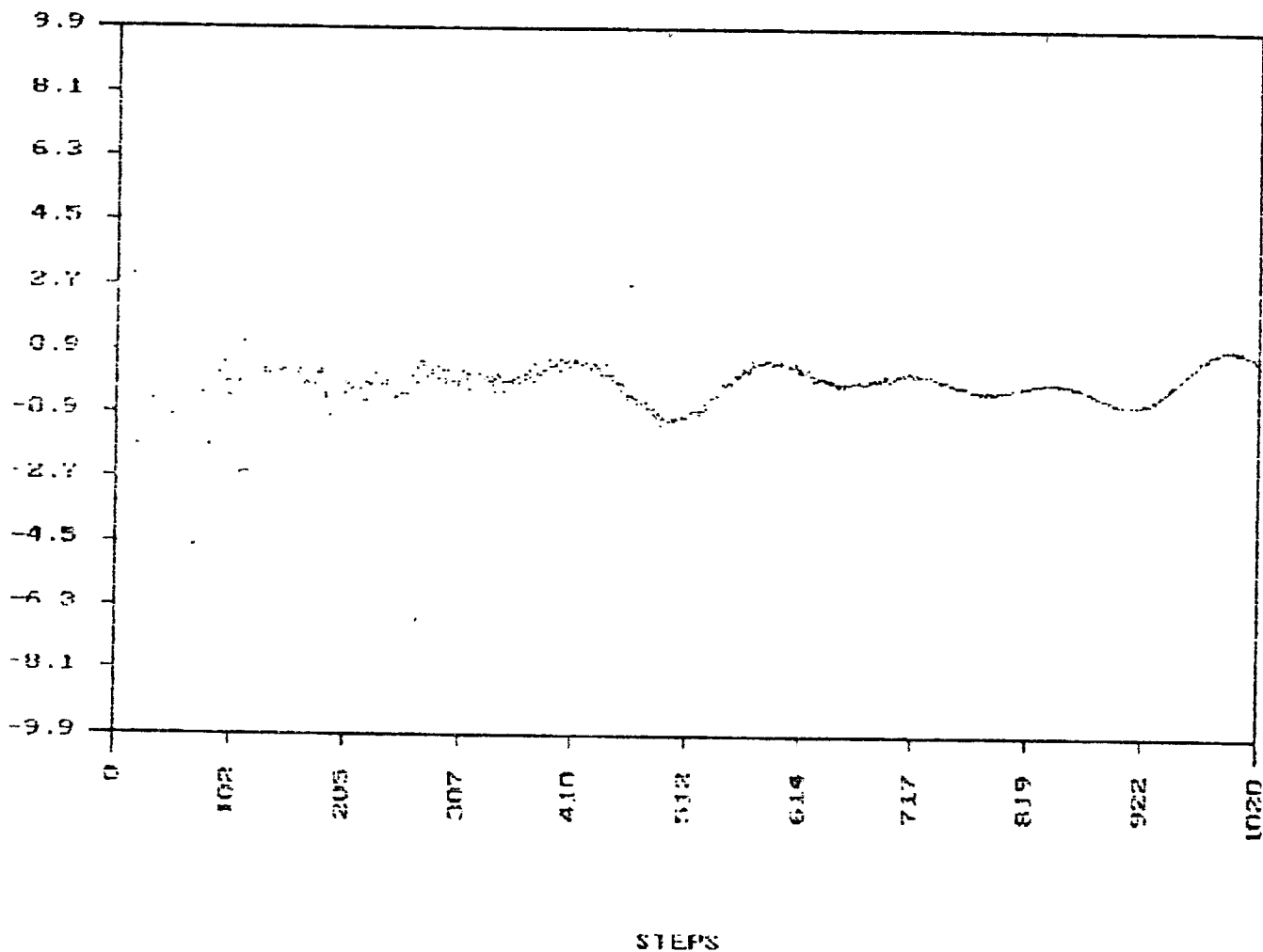


รูปที่ 5.31 สัญญาณเอาต์พุตจากวงจรถักสัญญาณรบกวน  $n_1/n_0 = 1$



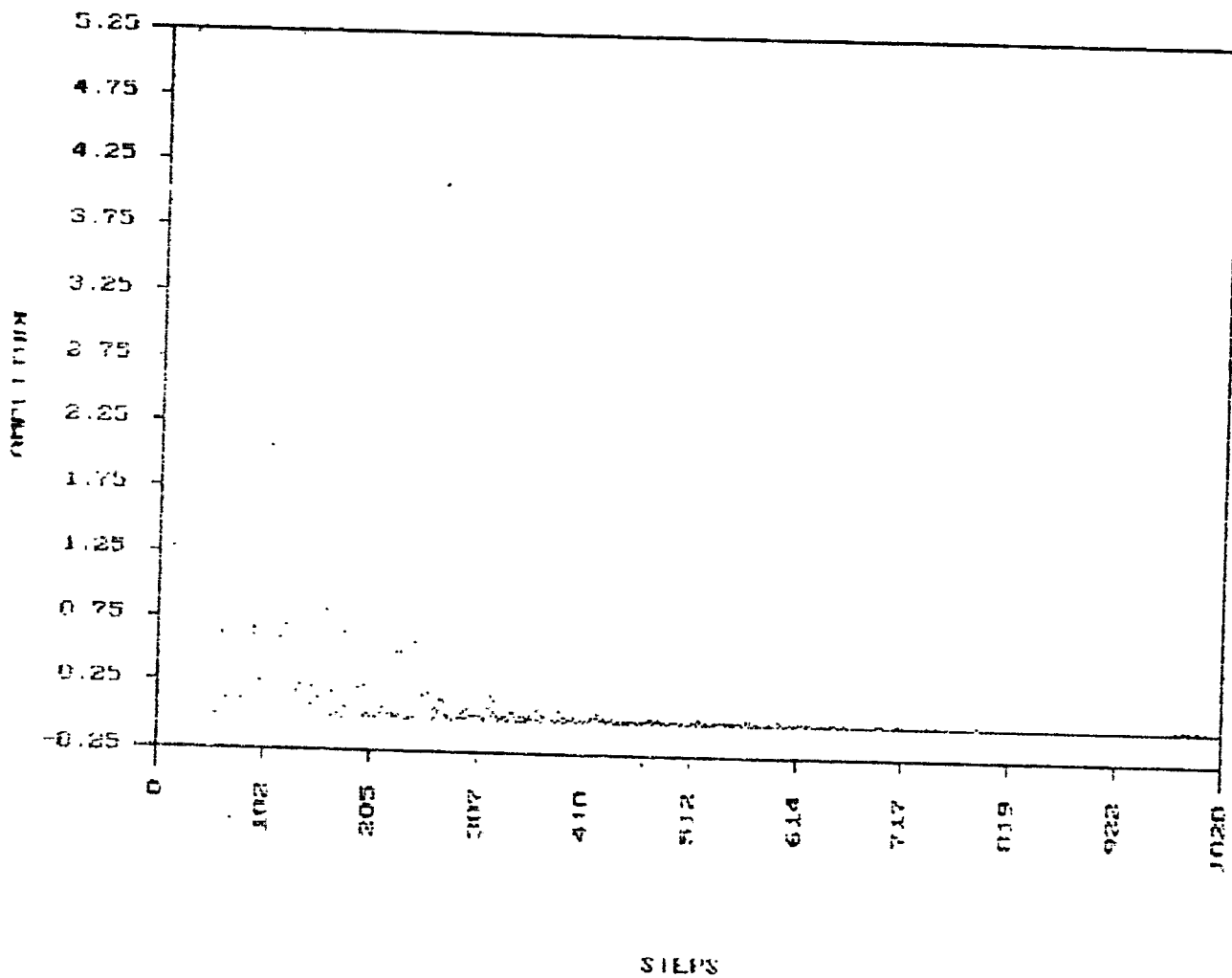
รูปที่ 5.32 กำลังงานของสัญญาณรบกวน  $n_1/n_0 = 1$

REFERENCE INPUT = 2



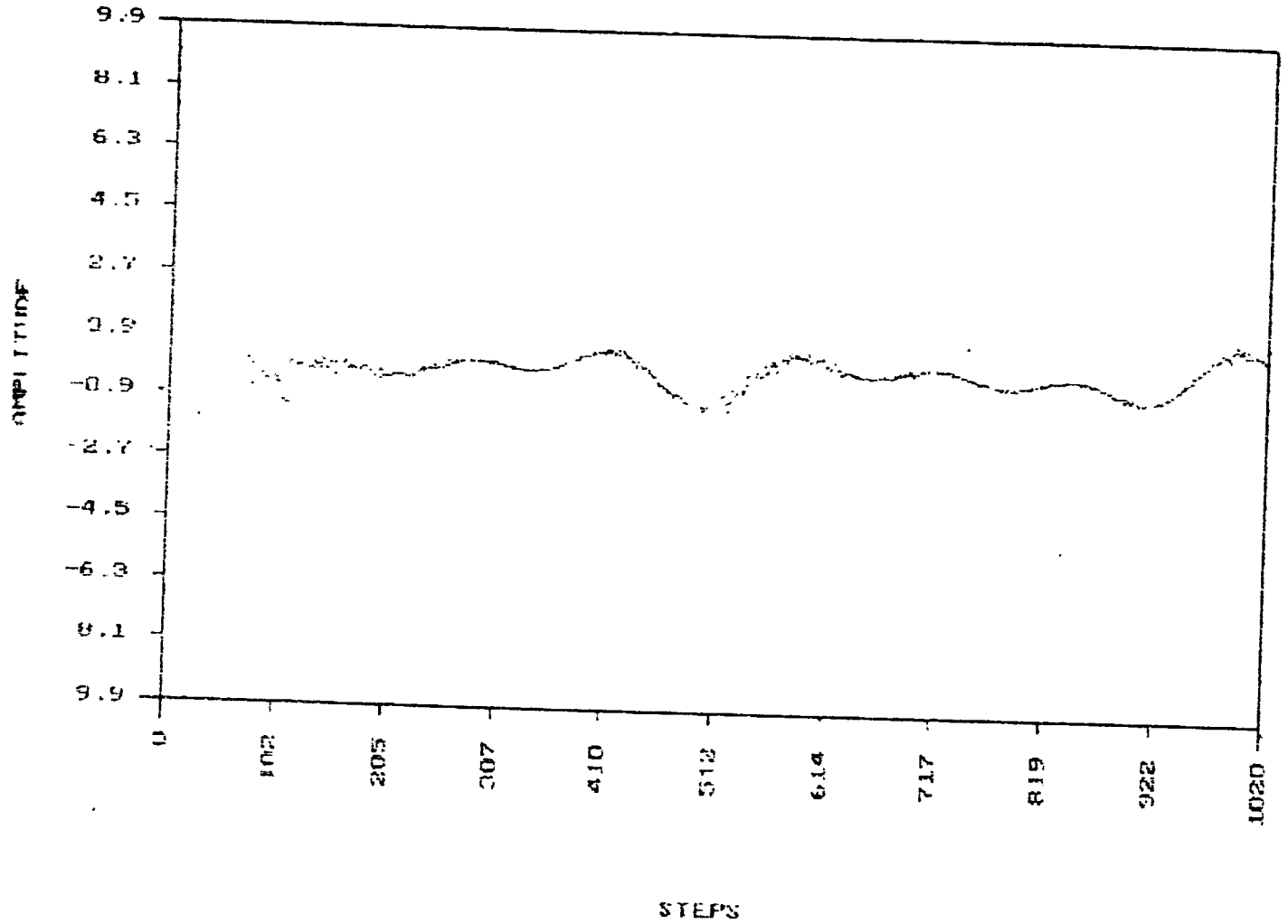
รูปที่ 5.33 สัญญาณเออร์เกทจากวงจรถ่ายสัญญาณรบกวน  $n_1/n_0 = 2$

MEAN SQUARE ERROR (N1=2)



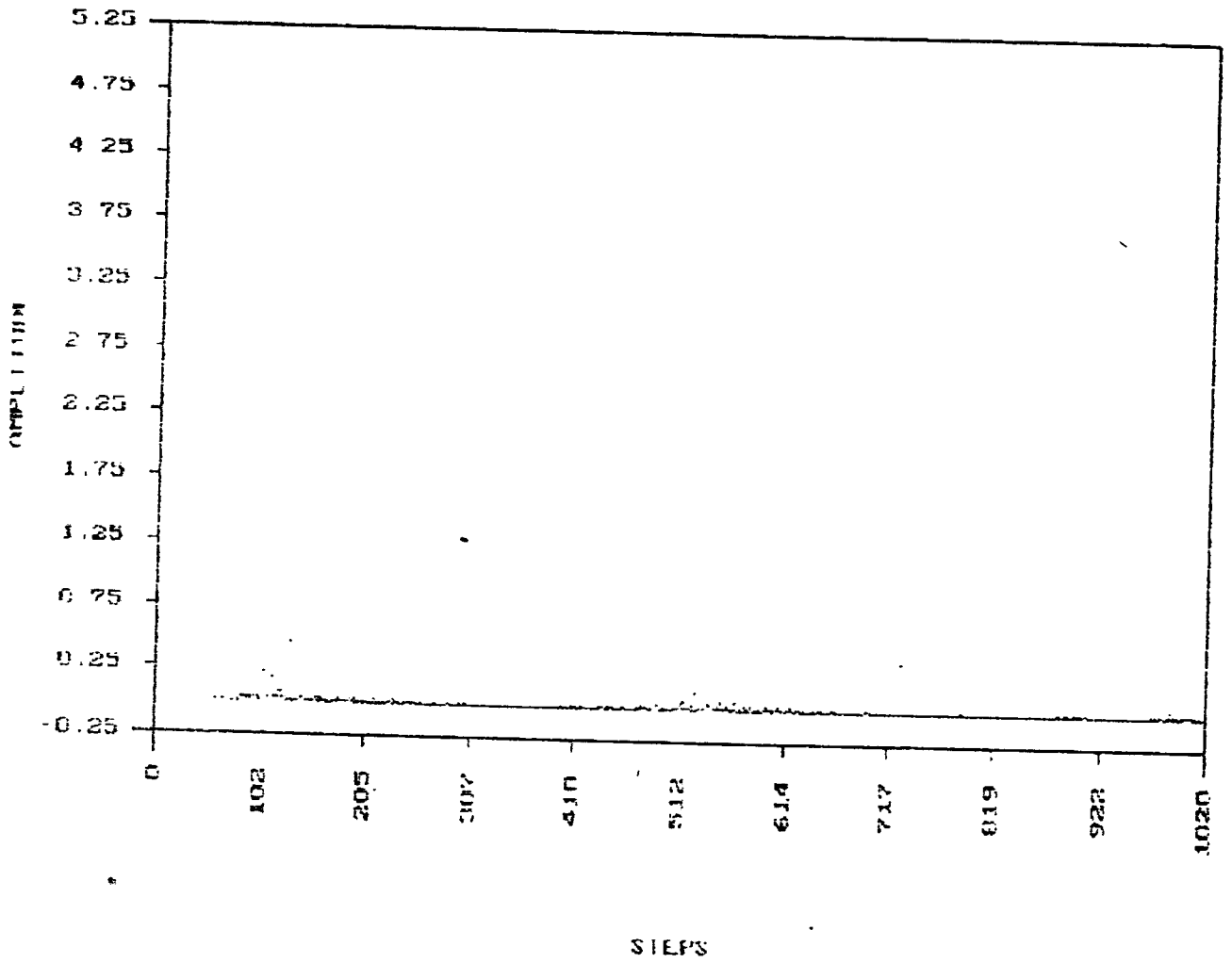
รูปที่ 5.34 กำลังงานของสัญญาณรบกวน  $n_1/n_0 = 2$

REFERENCE INPUT = 3



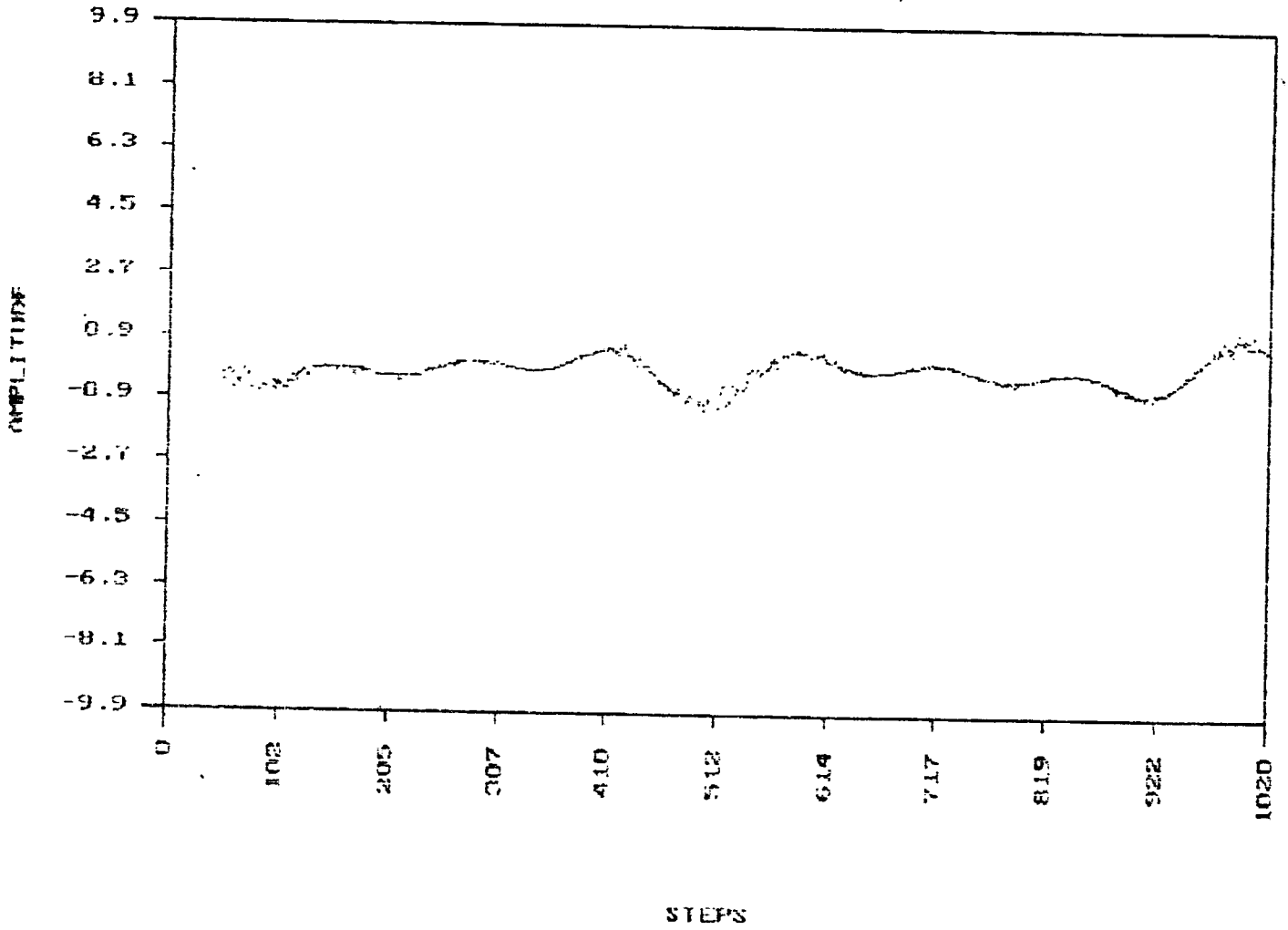
รูปที่ 5.35 สัญญาณเอาต์พุตจากวงจรถักำจัดสัญญาณรบกวน  $n_1/n_0 = 3$

MEAN SQUARE ERROR (N1=3)



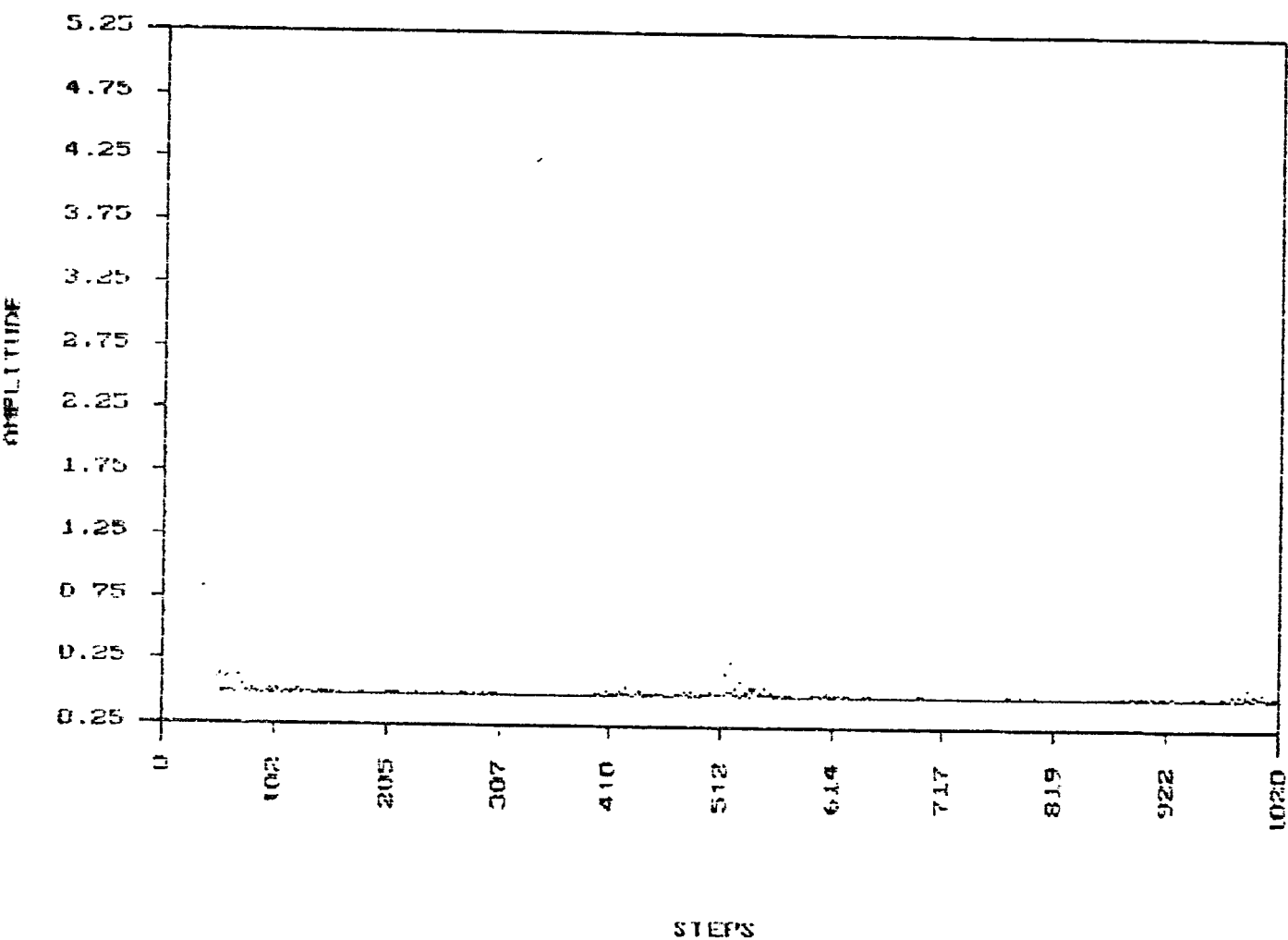
รูปที่ 5.36 กำลังงานของสัญญาณรบกวน  $n_1/n_0 = 3$

REFERENCE INPUT = 4



รูปที่ 5.37 สัญญาณเอาต์พุตจากวงจรกำจัดสัญญาณรบกวน  $n_1/n_0 = 4$

## MEAN SQUARE ERROR (N1=4)



รูปที่ 5.38 กำลังงานของสัญญาณรบกวน  $n_1/n_0 = 4$

### 5.2.3 การทดลองที่ 6 : ผลจากการแปรค่า สัญญาณรบกวน ( $n_0$ )

เงื่อนไขคงที่ทุกกรณี :

ขนาดสัญญาณอินพุตอ้างอิง  $n_1 = 3$

พารามิเตอร์ของการลู่เข้า  $\mu = 0.0008$

ตัวแปร : ขนาดของสัญญาณรบกวน  $n_0$  ในอินพุตหลัก

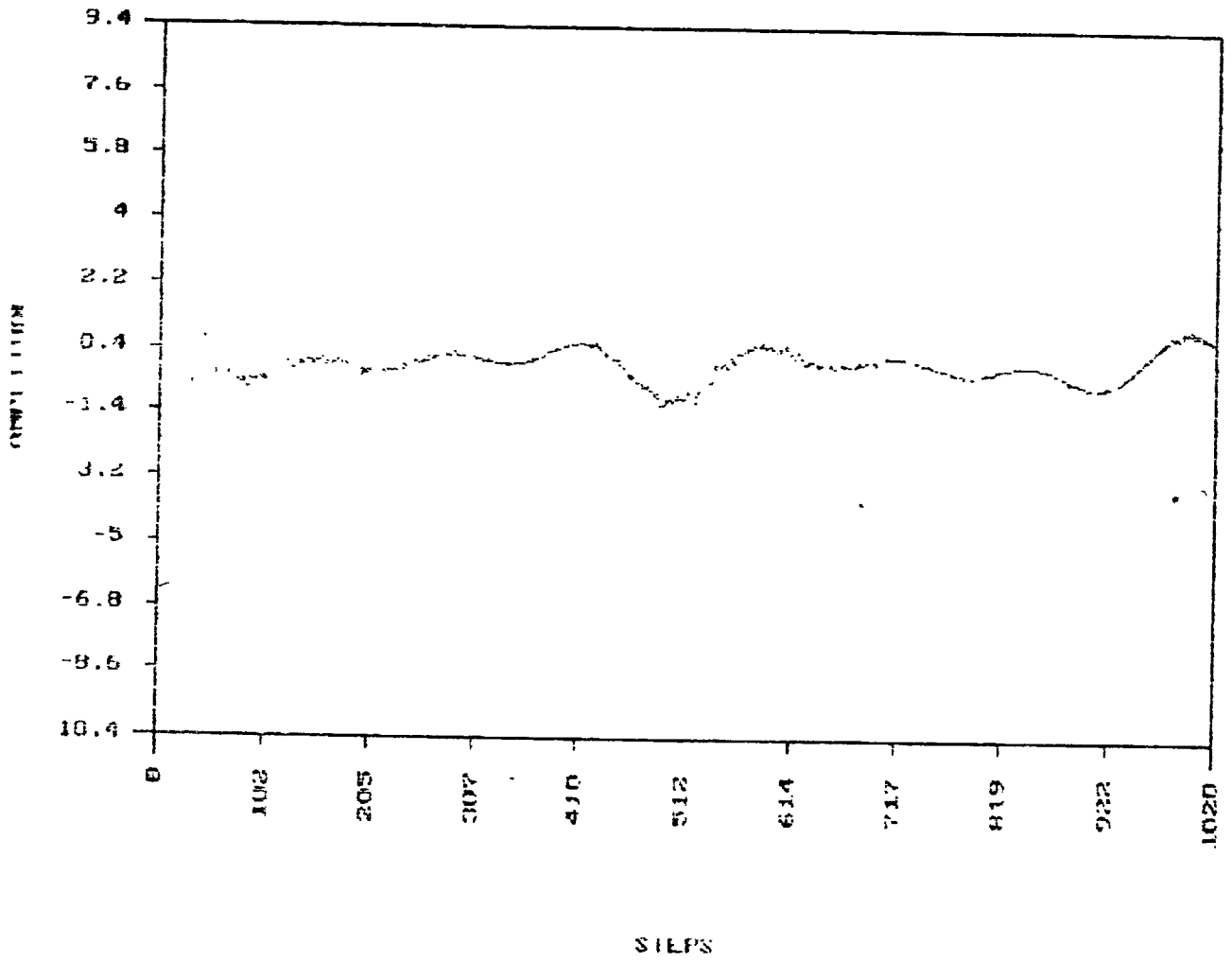
ได้ผลลัพธ์แสดงในรูปที่ 5.39 ถึงรูปที่ 5.44 และสรุปผลแสดงในตารางที่ 5.6

$n_0/s$	steps	S/N (db)
1	150	6.371
2	125	0.793
3	100	-2.604

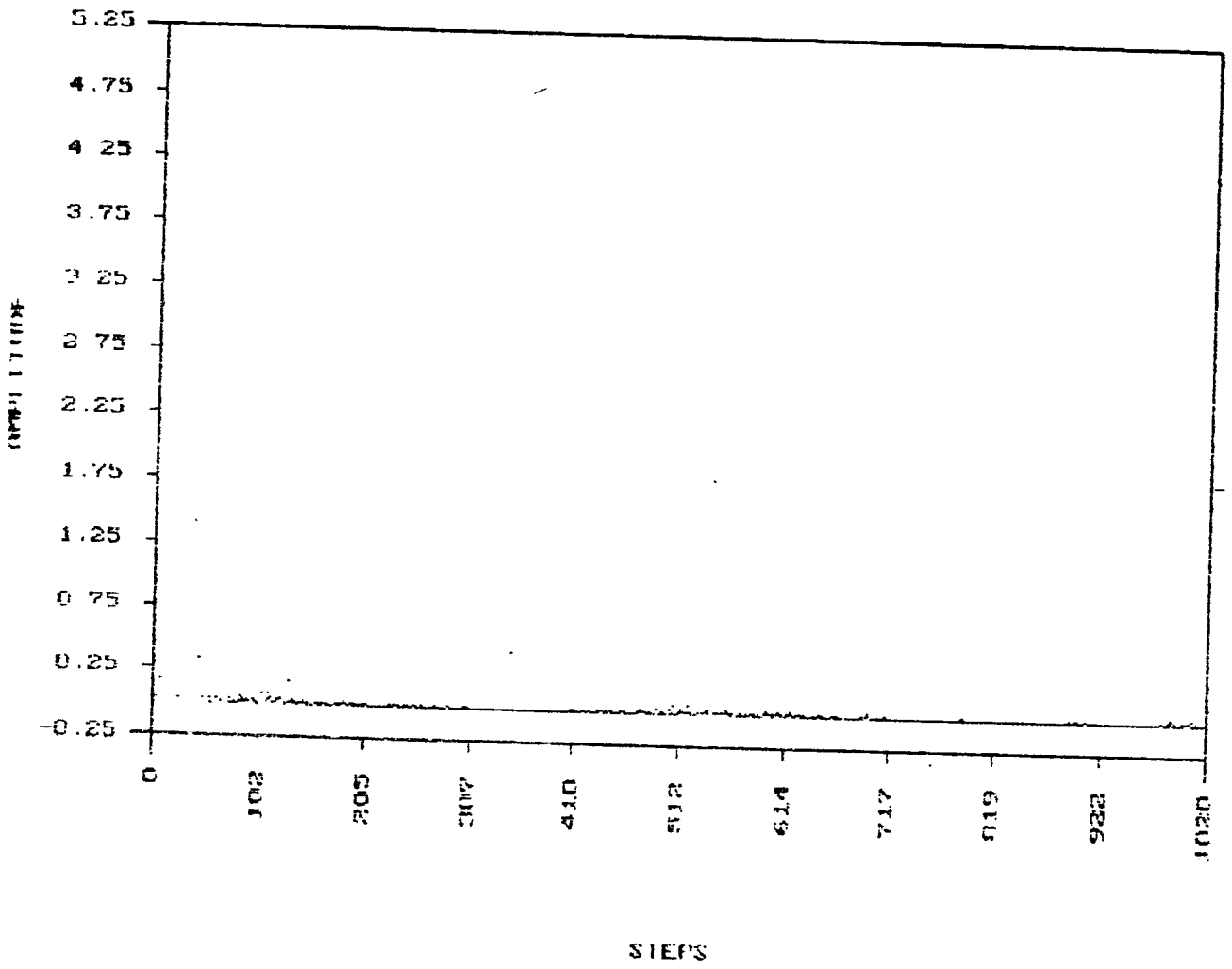
ตารางที่ 5.6

สรุปผลการทดลองได้ว่า ค่าพารามิเตอร์ของการลู่เข้า ที่มากมีผลทำให้ระบบเข้าสู่จุดที่ ค่าความผิดพลาด ลดลงเหลือ 1% ของค่าสูงสุด ได้อย่างรวดเร็ว และ ค่าS/N ก็จะมีมากขึ้นด้วย แต่ความผิดพลาดที่รูปสัญญาณก็จะปรากฏมากด้วย สัญญาณรบกวนในอินพุตอ้างอิง ( $n_1$ ) ควรจะมีค่า มากกว่า สัญญาณรบกวนในอินพุตหลัก ( $n_0$ ) และ ยิ่งอัตราส่วนของ S/N ของสัญญาณอินพุตหลัก ยิ่งมากเท่าไร วงจรก็จะสามารถทำงานได้ ดีขึ้น และรวดเร็วยิ่งขึ้น

NOISE/SIGNAL = 1

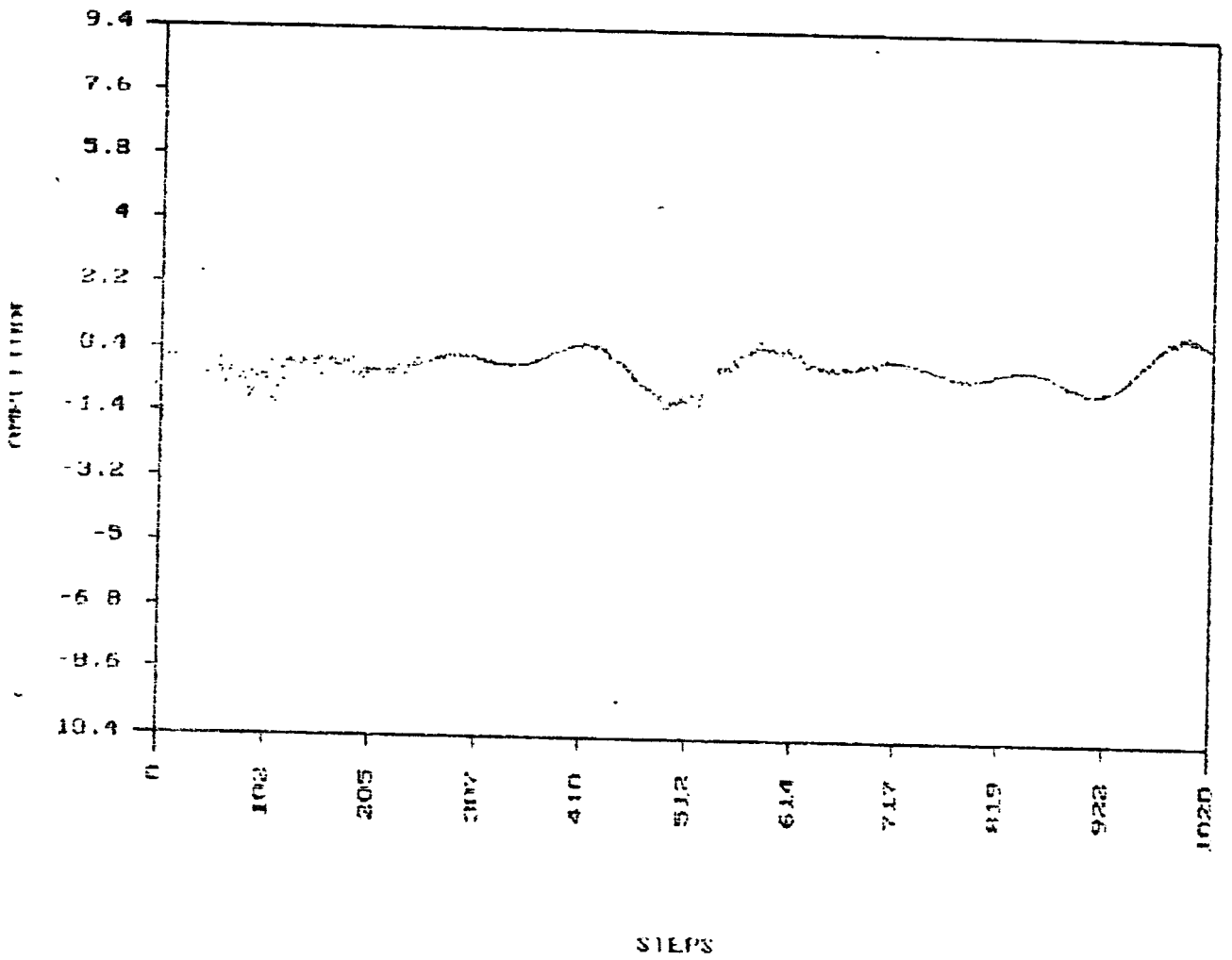


รูปที่ 5.39 สัญญาณเอากันที่พุดจากวงจรกำจัดสัญญาณรบกวน  $n_0/s = 1$



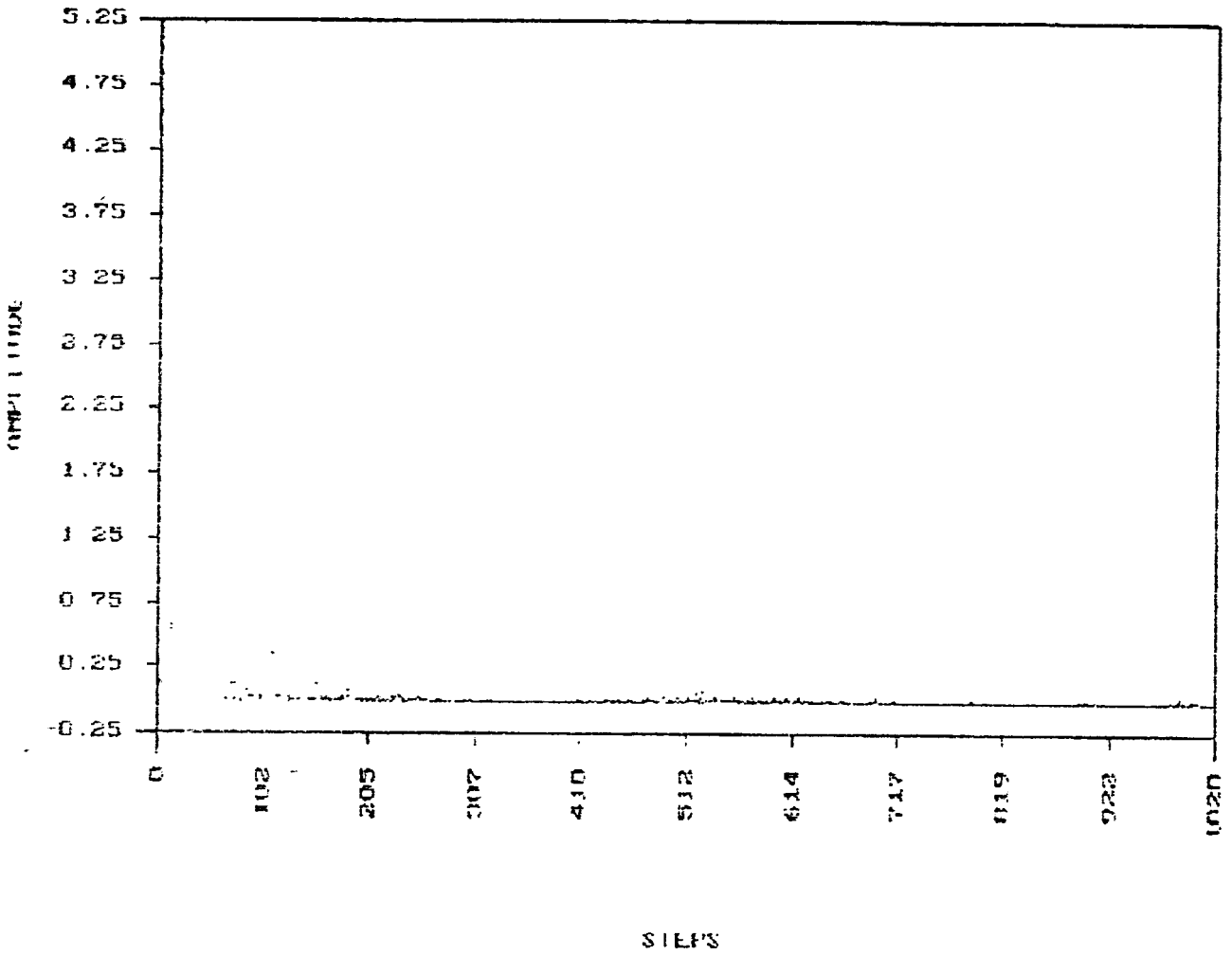
รูปที่ 5.40 กำลังงานของสัญญาณรบกวน  $n_0/s = 1$

NOISE/SIGNAL = 2



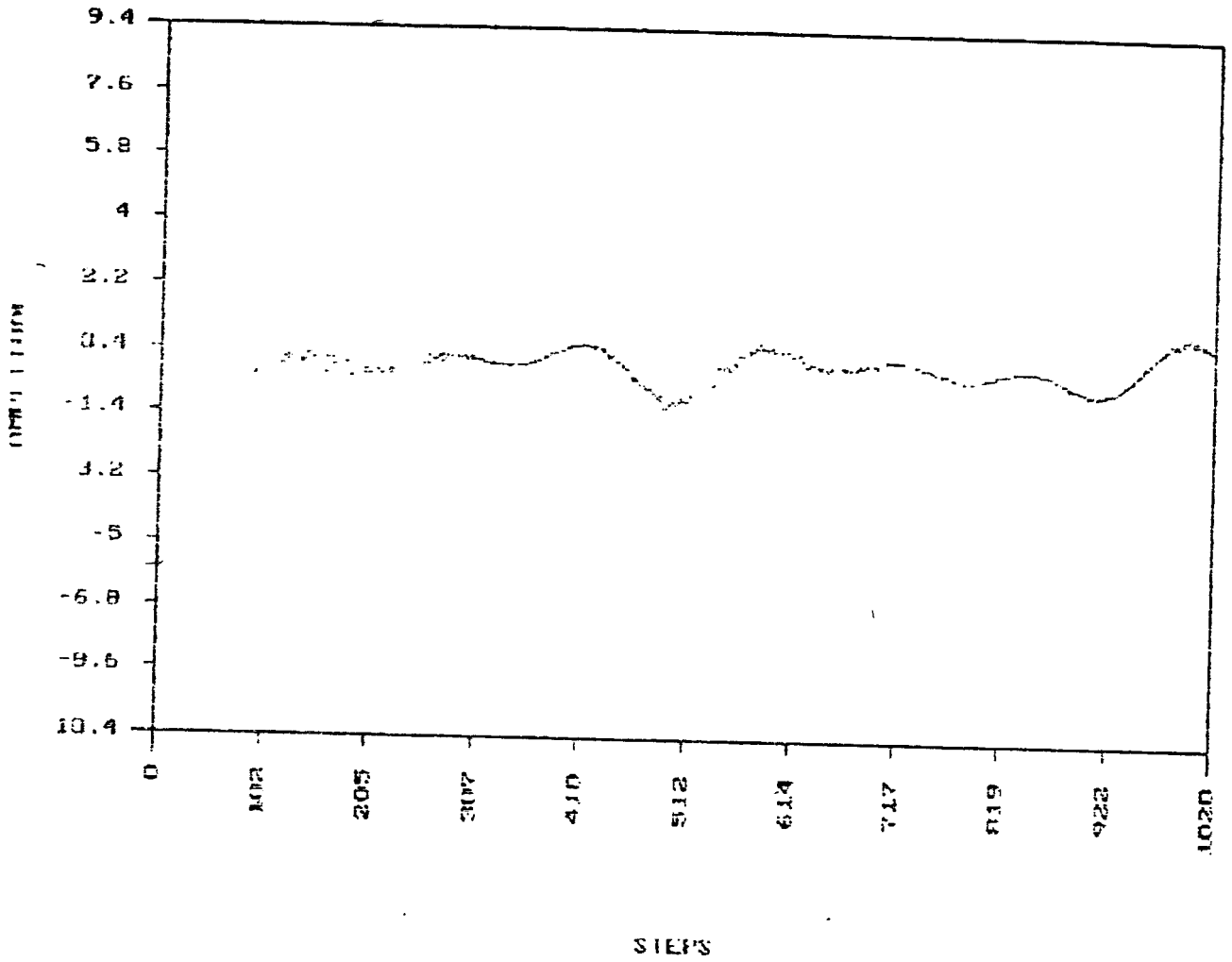
รูปที่ 5.41 สัญญาณเอาก้พุงจากวงจรถ้าจัดสัญญาณรบกวน  $n_0/s = 2$

MEAN SQUARE ERROR (N/S=2)



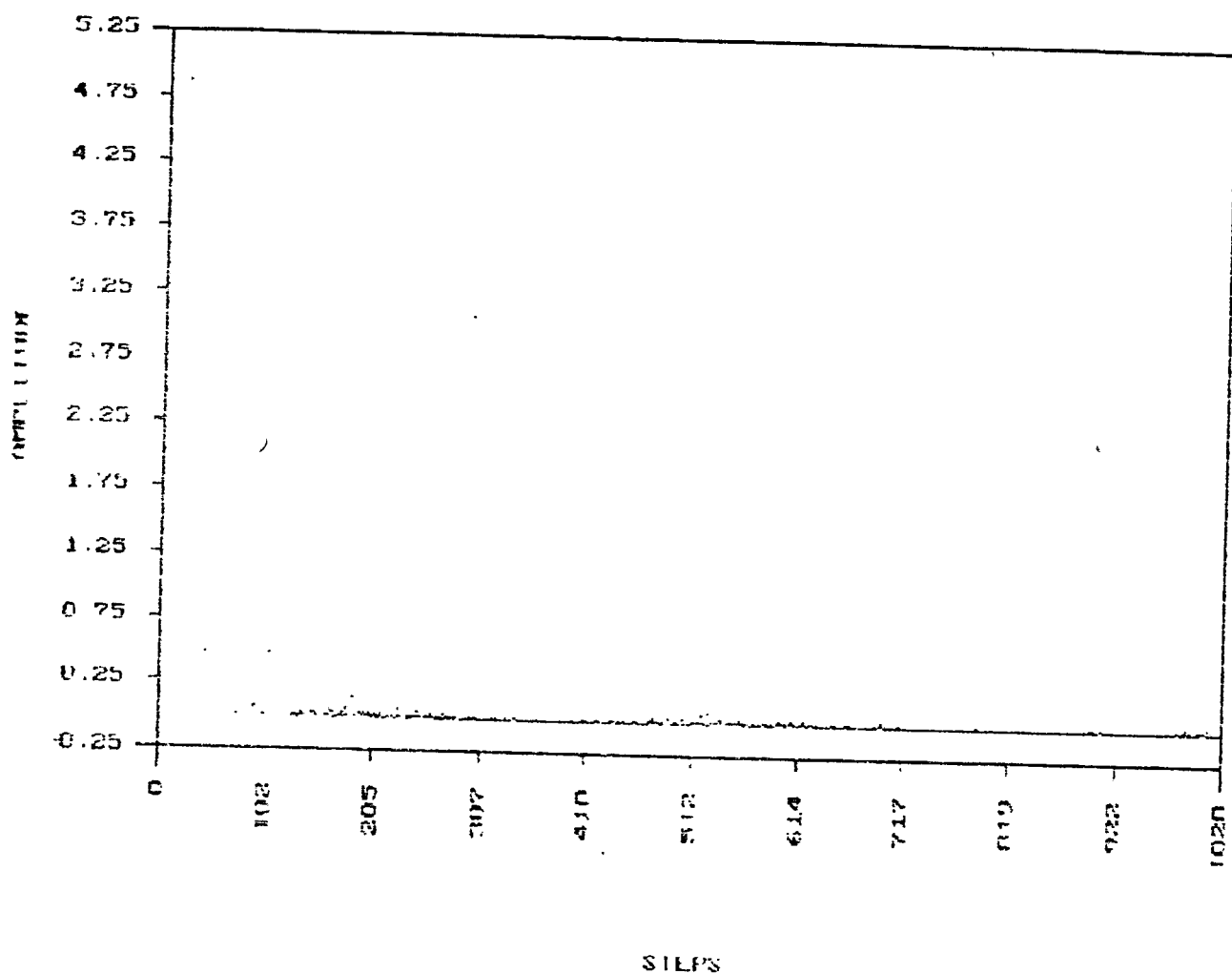
รูปที่ 5.42 กำลังงานของสัญญาณรบกวน  $n_0/s = 2$

NOISE/SIGNAL = 3



รูปที่ 5.43 สัญญาณเอาร์ทัพจากวงจรถ่ายสัญญาณรบกวน  $n_0/s = 3$

## MEAN SQUARE ERROR (N/S=3)



รูปที่ 5.44 กำลังงานของสัญญาณรบกวน  $n_0/s = 3$

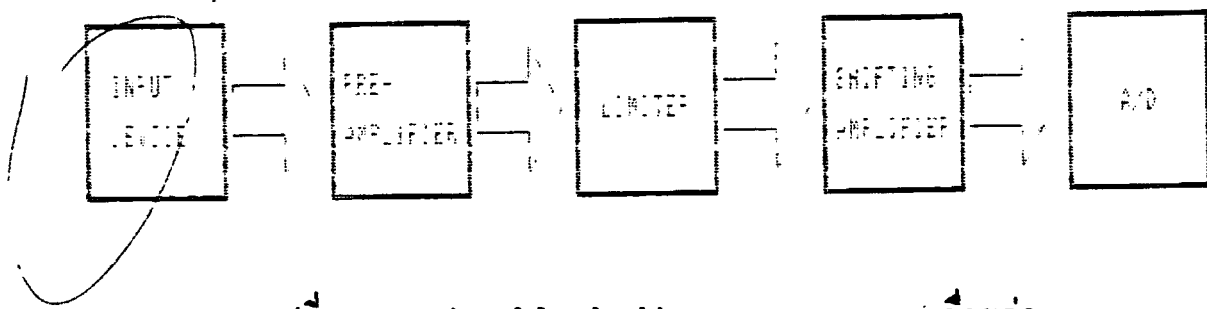
## บทที่ 6

### วงจรเชื่อมต่อ

ในบทที่ 5 เป็นเพียงการจำลองสัญญาณจาก ฟังก์ชันคณิตศาสตร์เท่านั้น ในการออกแบบใช้งานจริง ผู้ออกแบบสามารถนำสมการหลัก ของ LMS algorithm คือสมการที่ 3.15 ไปประยุกต์ใช้กับสัญญาณไฟฟ้า โดยให้เครื่องคอมพิวเตอร์อ่านสัญญาณอินพุตหลัก และสัญญาณอินพุตอ้างอิง ผ่านทางวงจรแปลงสัญญาณอนาล็อก เป็นสัญญาณดิจิทัล (A/D convertor) แล้วจึงทำการคำนวณด้วยเครื่องคอมพิวเตอร์ ส่วนสัญญาณเอาต์พุตที่ต้องการก็สามารถส่งออกจากคอมพิวเตอร์ผ่านทาง วงจรแปลงสัญญาณดิจิทัล เป็นสัญญาณอนาล็อก (D/A convertor)

สิ่งที่สำคัญเมื่อให้สัญญาณอนาล็อกผ่านวงจร A/D เข้ามาคือการจัดระดับสัญญาณให้เหมาะสมกับวงจร A/D ซึ่งวงจร A/D โดยทั่วไปจะรับสัญญาณอินพุตอยู่ในช่วงที่จำกัด เช่น 0 ถึง 5 โวลต์ หรือ 0 ถึง 10 โวลต์ ดังนั้นจึงต้องจัดวงจรขยายสัญญาณอินพุต ที่เป็นสัญญาณอนาล็อก ให้สามารถเชื่อมต่อ อุปกรณ์รับสัญญาณ (input sensor) เข้ากับวงจร A/D ได้

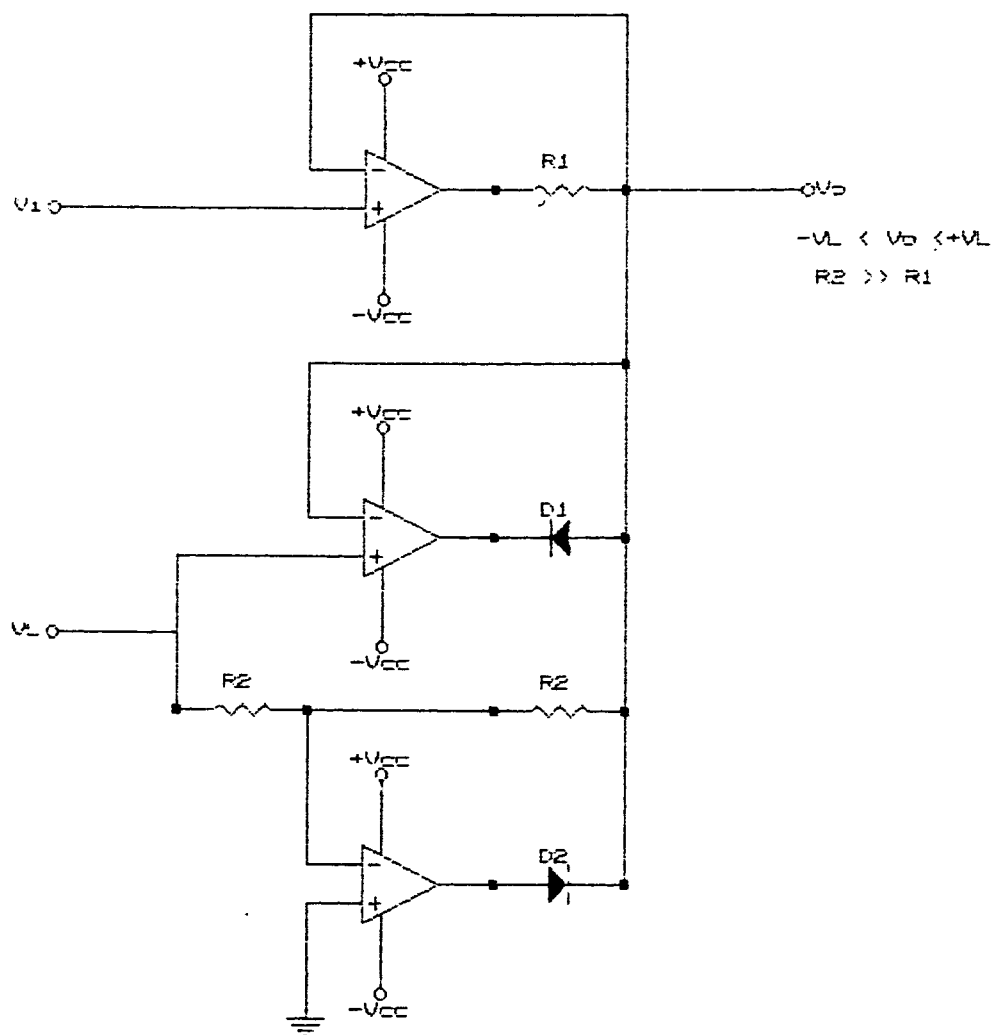
นอกจากนี้ จากบทที่ 4 สัญญาณทุกสัญญาณที่ปรากฏในการคำนวณ จะต้องมียุคเฉลี่ยเป็นศูนย์ (zero mean) ดังนั้นในบทนี้จะแสดงหลักการของวงจรเชื่อมต่อ อุปกรณ์รับสัญญาณกับวงจร A/D โดยมีส่วนประกอบ 2 ส่วนคือ ส่วนวงจรจำกัดขนาดสัญญาณ (clamer) และ วงจรเลื่อนสัญญาณ (shifting amplifier) โดยแสดงบล็อกไดอะแกรม การใช้วงจรทั้งสองในรูปที่ 6.1 เพื่อให้อุปกรณ์รับสัญญาณเชื่อมต่อกับวงจรได้อย่างเหมาะสม และมีคุณลักษณะที่ถูกต้องสำหรับข้อกำหนดของ LMS algorithm



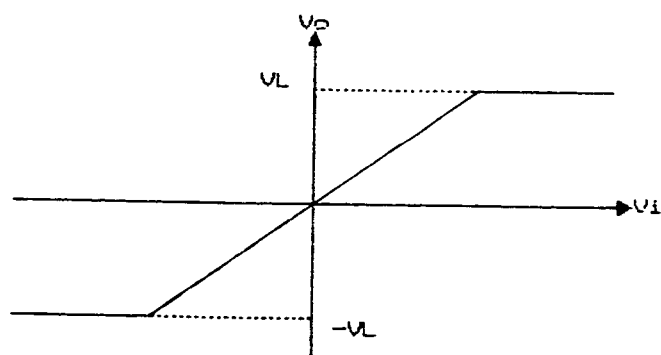
รูปที่ 6.1 แสดง block diagram ของวงจรเชื่อมต่อ

### 6.1 วงจรจำกัดขนาดสัญญาณ (Clamper)

ตามวงจรในรูปที่ 6.2 ให้อินพุทของวงจรจำกัดสัญญาณ ( $V_i$ ) มาจากภาคขยายสัญญาณ ของอุปกรณ์รับสัญญาณ และศักดาบอจุดจำกัด จะเข้าที่  $V_L$  คุณสมบัติของผลตอบสนองของวงจรนี้แสดงในรูปที่ 6.3



รูปที่ 6.2 วงจรจำกัดขนาดสัญญาณ



รูปที่ 6.3 คุณสมบัติของวงจรจำกัดสัญญาณ

การทำงานของวงจรมีสามารถอธิบายได้ดังนี้ ผลตอบสนองของวงจรตามรูปที่ 6.3 แยกได้เป็นส่วนต่าง ๆ 3 ส่วน คือ

1. ส่วนผลตอบสนองเชิงเส้น
2. ส่วนจำกัดศักดา้านบวก
3. ส่วนจำกัดศักดา้านลบ

ออปแอมป์ทั้งสามตัวจะทำหน้าที่ควบคุมสัญญาณ ในส่วนต่าง ๆ แยกกัน โดย ส่วนที่เป็นการตอบสนองเชิงเส้น (linear) จะผ่านออกไปที่เอาต์พุตโดยออปแอมป์ตัวบน จัดวงจรเป็น voltage follower เนื่องจาก ไดโอดทั้งสองไม่นำกระแส ทำให้ออปแอมป์ทั้งสองตัวว่าง ไม่มีผลต่อวงจร

เมื่อเอาต์พุต  $V_o$  ขึ้นถึงระดับ  $+V_L$  จะขับให้ออปแอมป์ตัวกลางซึ่งเป็นคอมพาราเตอร์ มีเอาต์พุตเป็นลบ (เพราะศักดาที่ขา inverting มากกว่าศักดาที่ขา noninverting) ทำให้ไดโอด  $D_1$  นำกระแส ทำให้ออปแอมป์ตัวนี้ถูกจัดวงจรเป็น voltage follower ทำให้  $V_o$  มีศักดาเท่ากับ  $+V_L$  ในขณะเดียวกัน ออปแอมป์ตัวบนจะถูกตัดออกจากการทำงาน (override) ด้วยเอาต์พุตส่วนที่มากกว่า  $+V_L$  ถูกดึงกระแสผ่าน  $R_1$  โดยออปแอมป์ตัวกลาง ทำให้เอาต์พุตรักษาศักดาที่  $+V_L$

เมื่อเอาต์พุต  $V_o$  แกว่งลงถึงระดับ  $-V_L$  ที่ขา inverting input ของออปแอมป์ตัวล่างจะมีศักดาเป็นศูนย์ ทำให้เอาต์พุตของออปแอมป์ตัวนี้เป็นบวก (เพราะศักดาที่ขา noninverting มากกว่าศักดาที่ขา inverting) ทำให้ไดโอด  $D_2$  นำกระแส เป็นการตัดออปแอมป์ตัวบนออกจากการทำงาน รักษาศักดาที่น้อยกว่า  $-V_L$  ที่  $-V_L$  เช่นเดียวกับลักษณะของการรักษาศักดา้านบวกที่ได้อธิบายมาแล้ว

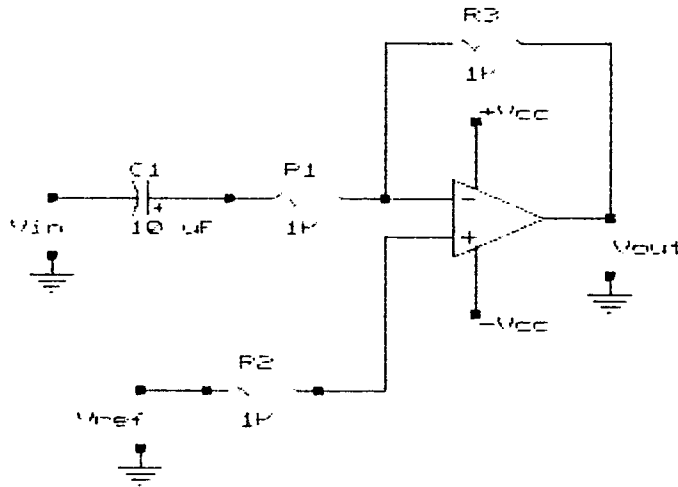
## 6.2 วงจรเลื่อนสัญญาณ (Shifting amplifier)

ตามรูปที่ 6.4 วงจรนี้มีหลักการ เป็นวงจรบวกสัญญาณแบบกลับเฟส (inverting summing amplifier) โดยบวกสัญญาณที่มาจากวงจรจำกัดสัญญาณ ( $V_{in}$ ) กับศักดาตรงอ้างอิง ( $V_{ref}$  DC) ได้เอาต์พุตเป็นผลบวกที่กลับเฟสสัญญาณ กับสัญญาณ  $V_{in}$  ซึ่งหาได้จากสมการในคที่ขา inverting ของออปแอมป์

$$\frac{V_{in}}{R_1} + \frac{V_{ref}}{R_2} + \frac{V_{out}}{R_3} = 0$$

เพราะฉะนั้น

$$V_{out} = -R3 \left[ \frac{V_{in}}{R1} + \frac{V_{ref}}{R2} \right]$$



รูปที่ 6.4 วงจรเลื่อนสัญญาณ

จากการใช้ทั้งสองวงจรมีเป็นวงจรเชื่อมต่อ ระหว่างอุปกรณ์รับสัญญาณ กับวงจร A/D ทำให้สามารถจัดให้สัญญาณอินพุท มีขนาดที่ไม่เกินข้อกำหนด ของ A /D และ วงจรเลื่อนสัญญาณจะเลื่อนศักดาของสัญญาณ เพื่อให้สามารถเชื่อมต่อกับ วงจร A/D ที่ไม่สามารถรับค่าศักดาที่เป็นลบได้ ซึ่งผลของการปรับสองวงจรมี จะทำให้สามารถรู้ค่าเฉลี่ยของสัญญาณที่เข้าทางอินพุท ของวงจร A/D ได้ ทำให้สามารถปรับแก้ LMS algorithm ให้ลบค่าเฉลี่ยของสัญญาณอันเนื่องมาจาก การจัดวงจร เพื่อเชื่อมต่อกับอุปกรณ์รับสัญญาณกับ A/D ก่อนที่จะนำสัญญาณที่อ่านเข้ามา ไปเข้าสมการ LMS algorithm

## บทที่ 7

### สรุปผลและวิจารณ์

จากการที่ได้ ศึกษาระบบปรับสถานะ โดยได้ประยุกต์ใช้ในการกำจัดสัญญาณรบกวน พบว่า ระบบปรับสถานะสามารถใช้งานได้ค่อนข้างรวดเร็ว โดยใช้สมการหลักเพียงสมการเดียว โดยที่ถึงแม้สภาวะรอบข้างของระบบ ซึ่งได้จำลองเป็นสัญญาณรบกวน มีการเปลี่ยนแปลงไป ระบบก็ยังสามารถทำงานได้

กรณีที่เป็นข้อเด่นของระบบปรับสถานะ คือ ถึงแม้สัญญาณที่ต้องการ จะผสมมาในปริมาณที่มาก ก็ยังสามารถดึงเอาข้อมูลที่ต้องการออกมาได้

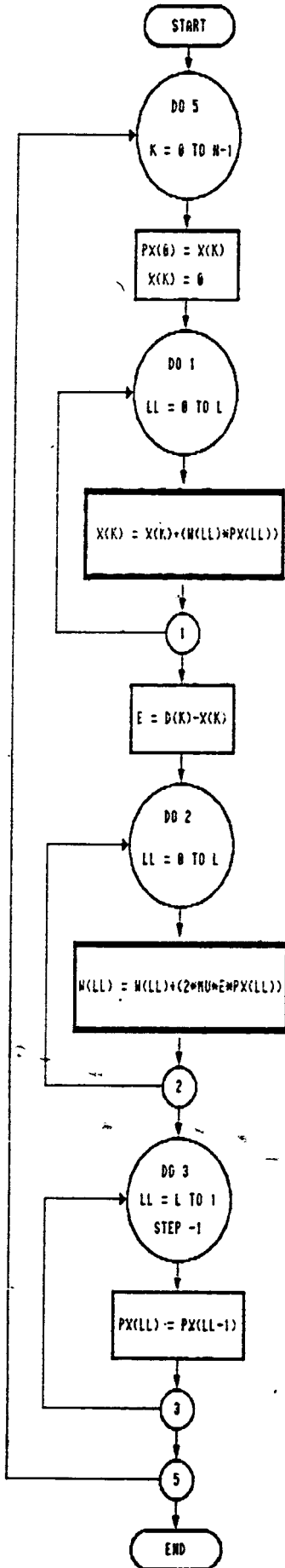
ปัญหาที่พบในการทำการจำลอง การทำงานของระบบกำจัดสัญญาณรบกวน ได้พบปัญหาคือ การใช้ LMS algorithm นั้นต้องมีการประมาณค่าแอมพลิจูด ซึ่งถ้าการประมาณค่านี้ทำให้เกิดความผิดพลาดมาก ประสิทธิภาพของการทำงานก็จะลดลง ดังนั้นถ้าสามารถทำการประมาณ ค่าแอมพลิจูด ที่ให้ผลลัพธ์ที่ใกล้เคียงกับ ค่าแอมพลิจูด ที่ถูกต้องได้ ระบบก็สามารถทำงานได้ถูกต้องและรวดเร็วยิ่งขึ้น

อีกปัญหาที่พบคือ ค่าพารามิเตอร์ของการลู่เข้า ไม่สามารถหาค่าที่เหมาะสมได้แน่นอน หาได้แต่ขอบเขตที่มากที่สุดเท่านั้น จึงต้องอาศัยการประมาณค่า

ระบบปรับสถานะนี้ ผู้สนใจสามารถศึกษาและพัฒนาต่อไปได้อีก ในหลาย ๆ ส่วน เช่น การประมาณค่าแอมพลิจูด หรือ การหาค่าพารามิเตอร์ของการลู่เข้าที่เหมาะสม หรือ การเชื่อมต่อในการใช้งานต่าง ๆ

ภาคผนวก

LMS ALGORITHM



N : LENGTH OF SIGNAL VECTOR  
 L : FILTER LENGTH  
 X(K) : REFERENCE INPUT VECTOR  
 D(K) : PRIMARY INPUT VECTOR  
 MU : CONDERGENCE PARATER

Subroutine for generating uniform noise

```
/******
```

uniform - generates zero mean uniform random number from -0.5 to 0.5

Returns one zero mean uniformly distributed random number as a double.  
No arguments, calls function rand() from the C library.

```
double uniform()
```

```
*****/
```

```
/* this define gives the biggest value rand() will return and  
is used only if not defined in a header file (non-ANSI  
implementations of C may require this) */
```

```
#ifndef RAND_MAX  
#define RAND_MAX 32767  
#endif
```

```
double uniform()  
{  
    return((double)(rand() & RAND_MAX) / RAND_MAX - 0.5);  
}
```

## Subroutine for generating Gaussian noise

```
/******
```

gaussian - generates zero mean unit variance Gaussian random numbers

Returns one zero mean unit variance Gaussian random numbers as a double.  
Uses the Box-Muller transformation of two uniform random numbers to  
Gaussian random numbers.

```
double gaussian()
```

```
*****/
```

```
double gaussian()
```

```
{
    static int ready = 0;          /* flag to indicated stored value */
    static double gstore;         /* place to store other value */
    double v1,v2,r,fac,gaus;
    double uniform();

/* make two numbers if none stored */
    if(ready == 0) {
        do {
            v1 = 2.*uniform();
            v2 = 2.*uniform();
            r = v1*v1 + v2*v2;
        } while(r > 1.0);          /* make radius less than 1 */

/* remap v1 and v2 to two Gaussian numbers */
        fac = sqrt(-2.*log(r)/r);
        gstore = v1*fac;          /* store one */
        gaus = v2*fac;           /* return one */
        ready = 1;               /* set ready flag */
    }

    else {
        ready = 0;               /* reset ready flag for next pair */
        gaus = gstore;          /* return the stored one */
    }

    return(gaus);
}
```

### หนังสืออ้างอิง

- [1] Widrow Bernard and Sterns Samuel "Adaptive Signal Processing," Prentice-Hall, Inc. Englewood Cliffs, NJ, USA, 1985
- [2] Sterns Samuel and Hush Don "Digital Signal Analysis," Prentice-Hall, Inc. Englewood Cliffs, NJ, Usa, 1990
- [3] Embree Paul and Kimble Bruce "C Language Algorithms for Digital Signal Processing," Prentice-Hall, Inc. Englewood Cliffs, NJ, USA, 1991
- [4] Stremler Ferrel "Introduction to Communication System," Addison-Wesley Publishing Company, London, 1977
- [5] Graeme Jerald "Designing With Operational Amplifiers," McGraw-Hill Book Company, New York, 1977

## กิตติกรรมประกาศ

ขอกราบขอบพระคุณอาจารย์ที่ปรึกษา ดร. วิวัฒน์ กิรานนท์ และ อ. โมนัย ไกรฤกษ์ เป็นอย่างมากที่ได้ช่วยให้ความรู้และคำแนะนำในการทำปริญญา  
นิพนธ์ฉบับนี้ และขอขอบคุณ อ. ปราโมทย์ วาดเขียน เป็นอย่างมากที่ช่วยให้ความรู้  
และคำแนะนำต่างๆ ตลอดจนให้ยืมอุปกรณ์ในการทำโครงงานนี้ ขอขอบคุณ  
ชุมชนคอมพิวเตอร์ ที่เอื้อเฟื้อโปรแกรม software ต่าง ๆ ตลอดจนเพื่อนๆ พี่ ๆ  
และน้อง ๆ ที่ได้ให้ความช่วยเหลือเป็นอย่างดี ทางผู้จัดทำจึงขอขอบพระคุณมา ณ ที่นี้